

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
GRADO DE INGENIERÍA INFORMÁTICA

APLICACIONES MÓVILES DISTRIBUIDAS PARA EL USO EN PROCESOS DE CODISEÑO

David de León Acosta

Tutor: Ignacio Aedo Cuevas

02/10/2014

Agradecimientos

Escribir estas líneas se me hace sumamente difícil por el hecho de saber que me va a faltar nombrar a gente que merece todo mi agradecimiento. A todos ellos, gracias de corazón.

A mi familia jamás le podré devolver una milésima parte de todo lo que han luchado para que yo pueda defender este Trabajo Fin de Grado. Se lo dedico a mis abuelos, a los que tuve la suerte de conocer y aprender de ellos. A mi padre, que me ha enseñado que en esta vida lo único que tenemos de valor es la palabra y que siendo una persona seria y responsable se puede llegar lejos. A mi madre, por hacerme saber que las decisiones eran mías y que ellos siempre estarán ahí para felicitarme en mis aciertos y ayudarme a levantar en mis tropiezos. A Jacob y Sandra, mis hermanos, porque sé que también han luchado mucho para que yo haya podido estudiar. A mis tíos y primos, con especial cariño a mi primo Dani por sacar tiempo para mí siempre que lo he necesitado. A los cuatro tesoros de la familia, mis sobrinos Adrián, Sergio, Daniela y Alejandra, que aunque me duela no verlos crecer, espero que algún día estén orgullosos de su tío.

Todo mi agradecimiento también para Gloria, por apoyarme y aguantarme durante estos últimos maravillosos años.

A Alexis Pulido, mi mejor amigo, el que siempre ha estado ahí cuando lo he necesitado sin importar el motivo. A Esteban Luis, porque a pesar de que hace mucho que no hablamos, me acuerdo a diario de toda lo que me ha ayudado. A Cristian García, con él empecé esta aventura y aunque nuestros caminos se separaron antes de lo que esperábamos, fue un placer compartir parte de este viaje.

A Yolanda y Jenifer, mis dos pequeñas grandes amigas, por darme esos momentos de locura o cordura justo cuando lo necesitaba. A Raúl Martín, por todas las horas que hemos pasado juntos, por el sufrimiento y las alegrías, espero poder seguir compartiendo momentos así, pero ahora, fuera de la Universidad. A Sergio Yuste, por su ayuda continua a cambio de nada. A Ana Lorente y Miguel, por el placer que ha sido conocerlos y trabajar con ellos durante este tiempo.

A mi tutor, Nacho, por haber depositado en mí su confianza. A Álvaro Montero por su paciencia y sus ganas de enseñar. A Sergio Herranz por escucharme y orientarme durante estos meses. Al resto de compañeros del laboratorio, por todos los momentos que pasamos juntos y por la ayuda y apoyo que me han dado.

Resumen

El presente trabajo fin de grado se desarrolla en el contexto del proyecto de investigación CoDICE (COdesigning Digital Cultural Encounters) orientado al desarrollo de objetos inteligentes mediante técnicas de codiseño para su uso en entornos culturales. El codiseño puede definirse como “el proceso de colaboración creativo entre diseñadores y personas no instruidas en diseño trabajan juntos en el proceso de desarrollo del diseño”. CoDICE nace como una aplicación web que ofrece a los usuarios las herramientas de codiseño para la creación de objetos inteligentes. La problemática del sistema ya creado reside en la limitación de tener que utilizar un ordenador para poder acceder al entorno de trabajo. Debido a ello, el objetivo de este proyecto es proporcionar a CoDICE una aplicación móvil multiplataforma que facilite la obtención y gestión de recursos multimedia desde el propio dispositivo. Además, se pretende ofrecer una solución que permita crear un entorno distribuido en tiempo real entre los distintos clientes del sistema. De esta manera se facilita el proceso de trabajo colaborativo para la creación de objetos inteligentes.

La solución desarrollada se caracteriza por el uso de tecnologías web que permiten crear aplicaciones móviles híbridas. Gracias a estas tecnologías se obtiene un equilibrio entre la facilidad de desarrollo que proporcionan y el alto rendimiento de las aplicaciones nativas. Para la creación del producto se ha empleado el patrón de diseño Modelo Vista Controlador, lo que permite conseguir una adaptación sencilla de la interfaz de la aplicación a los patrones de diseño de las distintas plataformas móviles para su posterior distribución. Mediante el uso de *websockets* se ha provisto al producto de un sistema de comunicación en tiempo real, apoyando así, el trabajo colaborativo. La solución ofrece la posibilidad de subir fotografías, videos y otros elementos multimedia al sistema, compartirlos y trabajar sobre ellos desde el propio dispositivo móvil.

La evaluación realizada ha demostrado que la solución cumple con los requisitos necesarios para ofrecer al usuario un entorno de trabajo móvil y distribuido en tiempo real, facilitando su uso en el entorno de trabajo orientado al diseño colaborativo ofrecido por CoDICE.

Abstract

This project is framed within the CoDICE research project. Its main objective is developing smart objects using co-design techniques for use in cultural encounters. Co-design can be defined as “the process of creative collaboration between designers and people, which have no training in design, in order to work together in the design development process”. CoDICE web application offers co-design tools for creating smart objects. The problem lies in the need of a personal computer in order to use the previous deployed application. The project goal is to create a hybrid mobile application for obtaining and managing multimedia resources. In addition, this solution provides a distributed real-time environment that facilitates the use of CoDICE in a collaborative environment.

Model View Controller pattern has been used for its development. This pattern allows us to design and develop interfaces for different mobile platforms easily. Also, we used websockets, it has provided a distributed real time environment thus supporting the collaborative work. The application allows users to upload, share and work with media files from mobile devices.

The evaluation has shown that the product provides the user with a mobile work environment distributed in real time. Also, it proved its proper integration in the collaborative design process of CoDICE.

Índice de contenidos

Agradecimientos	2
Resumen.....	3
Abstract	4
Índice de contenidos	5
Índice de ilustraciones.....	7
Índice de tablas	9
Glosario de términos	12
1 Introducción.....	14
1.1 Definición del problema	14
1.2 Objetivos	14
1.3 Fases del desarrollo.....	15
1.4 Estructura de la memoria.....	15
2 El estado de la cuestión	18
2.1 Revisión de procesos de codiseño	18
2.2 Revisión de arquitecturas móviles	25
2.3 Revisión de tecnologías móviles	30
2.4 Resumen del estado de la cuestión y elección de tecnologías revisadas	34
3 Gestión del proyecto software	36
3.1 Alcance del proyecto	36
3.2 Plan de trabajo	37
3.3 Estimación de recursos	39
3.4 Presupuesto	40
4 Solución.....	49
4.1 Descripción de la solución.....	49
4.2 El proceso de desarrollo.....	50
5 Evaluación	84
5.1 Proceso de evaluación.....	84
5.2 Análisis de resultados.....	87
6 Conclusiones	89
6.1 Aportaciones realizadas	89
6.2 Trabajos futuros	89

6.3	Problemas encontrados	90
6.4	Opiniones personales.....	91
7	Bibliografía	92
	Anexo I. Control de versiones	94
	Anexo II. Seguimiento de Trabajo Fin de Grado.....	96
	Anexo III. Especificación de requisitos	99
	Anexo IV. Especificación de casos de uso	107
	Anexo V. Especificación de casos de prueba.....	116
	Anexo VI. Prototipo final	125

Índice de ilustraciones

Ilustración 1. Diagrama de Afinidad	19
Ilustración 2. Group sketching	20
Ilustración 3. Mind map	21
Ilustración 4. Motivation matrix.....	22
Ilustración 5. Rough prototyping	23
Ilustración 6. Storyboard.....	24
Ilustración 7. Storytelling	25
Ilustración 8. Arquitectura aplicaciones móviles híbridas	27
Ilustración 9. Situación arquitecturas móviles	29
Ilustración 10. Ciclo de vida iterativo incremental.....	50
Ilustración 11. Arquitectura CoDICE.....	57
Ilustración 12. Alcance del sistema a implementar	58
Ilustración 13. Modelo Vista Controlador AngularJS	60
Ilustración 14. Encapsulado Apache Cordova	61
Ilustración 15. HUB SignalR	62
Ilustración 16. Subida de recursos mediante POST	63
Ilustración 17. Arquitectura del sistema	64
Ilustración 18. Mockups primera iteración – Login y Resources	75
Ilustración 19. Mockups primera iteración – Sign-Up, Pass Recovery y Add Resource	76
Ilustración 20. Mockups segunda iteración – Resource, My Account y left navigation	77
Ilustración 21. Mockups iteración – Comparativa de versiones para Resource	78
Ilustración 22. Pantalla de desarrollo de Intel® XDK.....	80

Ilustración 23. Estructura de archivos de AppFramework	81
Ilustración 24. Estructura de archivos del proyecto CoDICE	81
Ilustración 25. Archivos CSS	81
Ilustración 26. Archivos JavaScript	82
Ilustración 27. Pantalla de construcción de aplicación de Intel® XDK	83
Ilustración 28. Gantt - Planificación inicial del desarrollo	96
Ilustración 29. Gantt - Planificación inicial de la documentación	97
Ilustración 30. Gantt - Planificación final del desarrollo	97
Ilustración 31. Gantt - Planificación final de la documentación	98
Ilustración 32. Pantalla de login CoDICE versión móvil.....	125
Ilustración 33. Pantalla de alta de usuario CoDICE versión móvil.....	126
Ilustración 34. Pantalla de recuperación de contraseña CoDICE versión móvil	127
Ilustración 35. Pantalla de recursos CoDICE versión móvil	128
Ilustración 36. Pantalla de recurso individual CoDICE versión móvil.....	129
Ilustración 37. Pantalla de menú para añadir recursos CoDICE versión móvil	130
Ilustración 38. Pantalla de cuenta del usuario CoDICE versión móvil.....	131
Ilustración 39. Pantalla del menú lateral izquierdo desplegado CoDICE versión móvil.....	132
Ilustración 40. Pantalla de login CoDICE versión tablet	133
Ilustración 41. Pantalla de recursos CoDICE versión tablet	133
Ilustración 42. Pantalla de recurso CoDICE versión tablet.....	134
Ilustración 43. Pantalla de añadir recurso CoDICE versión tablet.....	134
Ilustración 44. Pantalla cuenta de usuario CoDICE versión tablet	135

Índice de tablas

Tabla 1. Características aplicaciones nativas	26
Tabla 2. Comparativa de arquitecturas	28
Tabla 3. Evaluación de arquitecturas	30
Tabla 4. Comparativa librerías para la creación de interfaces	32
Tabla 5. Evaluación librerías para creación de interfaces de usuario	32
Tabla 6. Comparativa tecnologías de conexión mediante <i>websockets</i>	34
Tabla 7. Resumen elección tecnologías	35
Tabla 8. Tareas asociadas a la creación del producto	38
Tabla 9. Tareas asociadas a la creación del documento	39
Tabla 10. Costes asociados a RRHH.....	40
Tabla 11. Bases de cotización contingencias comunes 2014	41
Tabla 12. Tipos de cotización 2014	41
Tabla 13. Tipos cotización por contratos de formación 2014	42
Tabla 14. Cuotas de cotización a la Seguridad Social.....	42
Tabla 15. Coste de personal	43
Tabla 16. Costes de hardware	44
Tabla 17. Costes de software	45
Tabla 18. Costes material fungible	45
Tabla 19. Costes indirectos	46
Tabla 20. Presupuesto por partidas	46
Tabla 21. Costes asociados al riesgo	47
Tabla 22. Beneficio	47

Tabla 23. Presupuesto detallado.....	48
Tabla 24. Método de pago	48
Tabla 25. Tabla de especificación de requisitos.....	51
Tabla 26. Tabla de especificación de casos de uso	54
Tabla 27. Matriz trazabilidad RF-CU.....	56
Tabla 28. Plantilla diseño vista	66
Tabla 29. Diseño vista login.....	67
Tabla 30. Diseño vista sign-up.....	67
Tabla 31. Diseño vista password recovery	67
Tabla 32. Diseño vista resources	68
Tabla 33. Diseño vista resource	68
Tabla 34. Diseño vista add resource	68
Tabla 35. Diseño vista new URI	69
Tabla 36. Diseño vista account.....	69
Tabla 37. Diseño vista left navigation	70
Tabla 38. Plantilla diseño controlador	70
Tabla 39. Diseño controlador LoginController	71
Tabla 40. Diseño controlador SignUpController	71
Tabla 41. Diseño controlador ForgotPasswordController.....	71
Tabla 42. Diseño controlador ResourcesController	72
Tabla 43. Diseño controlador ResourceController.....	73
Tabla 44. Diseño controlador AddResourcesController.....	74
Tabla 45. Diseño controlador AccountController	74
Tabla 46. Diseño controlador SlideController	74

Tabla 47. Plantilla para casos de prueba.....	84
Tabla 48. Matriz de trazabilidad CU-CPR	86
Tabla 49. Análisis de resultados de los casos de prueba	87
Tabla 50. Control de versiones.....	95

Glosario de términos

Aplicación distribuida. Aplicación que cuenta con distintos componentes que se ejecutan en entornos separados conectados mediante una red.

ASP.NET. Framework propietario de Microsoft para desarrollar aplicaciones web.

Design responsive. Filosofía de diseño web cuyo objetivo es adaptar la visualización de las páginas web a las características del dispositivo.

Framework. Ofrecen una funcionalidad definida utilizando patrones de diseño que facilitan el desarrollo de sistemas software.

Internet Information Services. Servidor web propietario de Microsoft.

jQuery. Biblioteca de JavaScript de código abierto que simplifica la manera de trabajar en el desarrollo web con respecto a uso directo de JavaScript.

Mockup. Modelo o diseño previo no funcional de una aplicación utilizado para la demostración o evaluación del diseño.

NHibernate. Herramienta de mapeo objeto-relacional para desarrollos .NET.

Plugin. Complemento que proporciona una nueva función a un sistema. En este documento está orientado a nuevas funciones de desarrollo.

POST. Método de petición HTTP. Envía los datos para que sean procesados por el recurso identificado incluido en el cuerpo de la petición. Resulta útil en la creación de un nuevo recurso o la actualización de los mismos.

Recurso. En el proyecto supone cualquier producto multimedia con el que el usuario pueda trabajar.

Responsive design. Filosofía de diseño web cuyo objetivo es adaptar la interfaz de las páginas web al dispositivo que se esté utilizando para su visualización.

Silverlight. Biblioteca de desarrollo para la creación de experiencias de usuario interactivas para aplicaciones web y móviles basado en la tecnología .NET de Microsoft.

Objetos inteligentes (*Smart objects*). Dentro del contexto del proyecto, se presentan como objetos o elementos tangibles a los que se les aplica tecnología digital para apoyar y mejorar su interacción con las personas.

Websockets. Tecnología que proporciona un canal de comunicación full dúplex y bidireccional sobre un único socket TCP.

1 Introducción

El codiseño supone el proceso de colaboración creativa entre diseñadores y personas no instruidas que trabajan juntos en el proceso de desarrollo del diseño [1]. A lo largo de este documento se presenta la creación de una aplicación móvil distribuida para el uso en procesos de codiseño dentro del contexto del proyecto CoDICE (COdesigning Digital Cultural Encounters).

1.1 Definición del problema

El proyecto europeo meSch nace con el objetivo de codiseñar nuevas plataformas que permitan crear exposiciones tangibles. De este modo, se pretende ofrecer nuevas experiencias interactivas mediante el uso de objetos inteligentes [2]. Dentro de este proyecto, un equipo de investigación de la Universidad Carlos III de Madrid está desarrollando una serie de herramientas software para el diseño de elaboración de estos objetos inteligentes [3]. En concreto, la versión actual de CoDICE comprende una aplicación web en la que los usuarios pueden compartir recursos y trabajar con ellos aplicando técnicas de codiseño.

El punto débil del sistema actual, es que requiere de un ordenador personal para alojar en la aplicación nuevos recursos multimedia como fotografías o vídeos sobre los que trabajar. Por ello, es necesario considerar una alternativa que permita a los usuarios añadir esos nuevos recursos multimedia al sistema de un modo rápido y eficaz desde cualquier lugar.

1.2 Objetivos

A lo largo de este apartado se recogen los objetivos a conseguir con el desarrollo de este proyecto. La definición de estos objetivos permitirá establecer las condiciones de validación para el producto desarrollado.

- **Creación de una aplicación móvil con capacidades multiplataforma.** La aplicación creada debe ser fácilmente portable a otros sistemas móviles. Requiere de una fácil adaptación para la creación de nuevas interfaces adaptadas a cada una de las plataformas móviles.
- **Obtención de recursos en tiempo real.** La aplicación debe ofrecer la capacidad de obtención de recursos en tiempo real para ser utilizado en actividades de diseño colaborativo.

- **Visualización y gestión de recurso en tiempo real.** La aplicación debe ofrecer al usuario la posibilidad de visualizar y trabajar con los recursos que se encuentran alojados en el sistema.

1.3 Fases del desarrollo

En este apartado se presentan las cuatro fases en los que está dividido el desarrollo de este proyecto.

- **Situación actual y estudio del problema.** El sistema a desarrollar forma de parte de un proyecto que cuenta con una aplicación web en fase de desarrollo avanzada. Por ello, en la primera fase se realiza un estudio de la aplicación web y se obtiene las necesidades que debe cubrir el nuevo producto a implementar. A partir de esas necesidades, se comienza con la búsqueda de alternativas tecnológicas que permitan definir la viabilidad del sistema.
- **Solución.** Esta fase se centra en la creación de un sistema que cubra las necesidades expuestas en el estudio del problema. Incluye tres etapas: **análisis**, **diseño** e **implementación**.

Durante la etapa de **análisis** se definen los fallos detectados en el estudio de viabilidad del sistema mediante la definición de requisitos. La detección de fallos durante esta etapa supone mejorar los índices de validación y verificación del producto.

A lo largo del **diseño** se presenta la arquitectura y las interfaces del sistema. Esta etapa define lo que se va a desarrollar durante la etapa de implementación de la solución.

La etapa de **implementación** define el proceso de desarrollo de software y el despliegue de la aplicación.

- **Evaluación.** La fase de evaluación permite demostrar la validez de la solución desarrollada.

1.4 Estructura de la memoria

El presente documento recoge la memoria del Trabajo Fin de Grado en Ingeniería Informática titulado “Aplicaciones móviles distribuidas para el uso en procesos de codiseño”. Se encuentra estructurado en siete capítulos y seis anexos:

- **Capítulo 1: Introducción.** A lo largo de este capítulo se recoge una visión general del proyecto, el problema planteado, los objetivos a conseguir con cada una de sus fases y la estructura de la documentación.
- **Capítulo 2: Estado de la cuestión.** En este capítulo y con el objetivo de contextualizar el proyecto se comienza con una revisión de los diferentes procesos de codiseño. A continuación, se realiza una revisión de las posibles arquitecturas y tecnologías a utilizar para la creación del producto presentando una comparación entre ellas. Para finalizar el estado de la cuestión, se expone un pequeño resumen en el que se muestran las tecnologías con las que se va a abordar el proyecto.
- **Capítulo 3. Gestión del proyecto software.** En esta sección del documento se presentan las estimaciones, planificaciones y costes asociados al proyecto.
- **Capítulo 4. Solución.** A lo largo de este capítulo se expone la solución del problema. Para ello se comienza con una descripción general de la solución. A continuación, se detalla el proceso de desarrollo dividido en cuatro partes fundamentales: modelo del proceso, análisis, diseño e implementación.
- **Capítulo 5. Evaluación.** En este apartado se define el proceso de evaluación del sistema. Contiene la creación de un plan de pruebas, la definición de los casos de pruebas y el análisis de los resultados obtenidos.
- **Capítulo 6. Conclusión.** En este capítulo se exponen las conclusiones del proyecto. Estas conclusiones incluyen las aportaciones realizadas, los problemas encontrados, una recomendación sobre trabajos futuros asociados al proyecto y la opinión personal del autor.
- **Capítulo 7. Bibliografía.** Se incluyen las referencias utilizadas para la redacción de este documento siguiendo el formato de cita proporcionado por el IEEE¹.
- **Anexo I. Control de versiones.** En el primer anexo se recoge el control de versiones de este documento. Contiene la versión del documento, la fecha y una pequeña descripción.
- **Anexo II. Seguimiento de Trabajo Fin de Grado.** En este anexo se encuentra el seguimiento del proyecto. Se presenta la planificación inicial y la planificación final del proyecto.

¹ <http://www.ieee.org/documents/ieeecitationref.pdf>

- **Anexo III. Especificación de requisitos.** A lo largo de este anexo se presenta la especificación de requisitos del proyecto.
- **Anexo IV. Especificación de casos de uso.** En este anexo se especifican los casos de uso del proyecto.
- **Anexo V. Especificación de casos de prueba.** A lo largo de este anexo se muestran las tablas asociadas a los casos de prueba diseñados para el proyecto.
- **Anexo VI. Prototipo final.** En el último anexo se incluyen las vistas del prototipo final entregado al cliente.

2 El estado de la cuestión

En el campo del diseño existe una corriente de investigación cada vez más extendida, orientada hacia un enfoque centrado en el usuario. A grandes rasgos, se puede afirmar que este nuevo enfoque está cambiando el panorama de la práctica del diseño, creando nuevos dominios de la creatividad colectiva [4]. Haciendo uso de estos nuevos dominios, se está trabajando en el proyecto CoDICE (COdesigning Digital Cultural Encounters), el cual tiene como objetivo la creación de objetos inteligentes para encuentros culturales mediante técnicas de codiseño.

Esta sección tiene como objetivo realizar un estudio que permita definir las bases para el desarrollo del trabajo enfocado en el contexto de CoDICE. Para ello, se comienza con la revisión de los distintos procesos de codiseño existentes. A continuación, se presenta una revisión de tecnologías móviles con el objetivo de determinar la mejor opción para desarrollar la solución propuesta en este trabajo.

2.1 Revisión de procesos de codiseño

Los procesos de codiseño suponen la aplicación de la creatividad colectiva en todo el proceso de diseño. De forma más concreta, se puede tomar como definición de codiseño como el proceso de colaboración creativa entre diseñadores y personas no instruidas en diseño que trabajan juntos en el proceso de desarrollo del diseño [1].

A continuación, se expone la revisión de las principales técnicas de codiseño.

Affinity diagrams

Los *affinity diagrams* (diagramas de afinidad) se presentan como un proceso en el que se sintetizan un conjunto de datos e ideas agrupándolas mediante las correlaciones naturales que existen entre ellas. Este tipo de diagramas fue creado por Kawakita Jiro, siendo también conocido como el método KJ [5].

Su uso está indicado cuando se pretende organizar un amplio conjunto de datos en el contexto de un problema complejo o que parece estar desorganizado, o cuando se requiere de la participación de todo el equipo.

Para utilizar esta técnica de codiseño, se debe comenzar con el planteamiento de un problema a un grupo de personas. Tras el planteamiento del problema se pasa a recoger las ideas que obtienen de manera individual cada uno de los miembros, teniendo en cuenta su visión personal del problema. Las ideas se deben plasmar en las tarjetas de forma simple y concreta ya que se busca que de un solo vistazo el resto de componentes del grupo puedan comprender lo que se propone. El siguiente paso es la clasificación de las tarjetas. Este proceso se lleva a cabo con el consenso de todos los componentes del grupo, se disponen todas las ideas sobre un tablero o una pared de manera que entre todos puedan agrupar por similitud las tarjetas dispuestas. Todos los grupos generados deberán contar con un título que será designado de manera grupal. Para finalizar con esta técnica, los miembros del grupo deben dibujar el diagrama y discutir sobre la solución del mismo. Con este fin, pueden crear una matriz en la que se enfrente la solución obtenida a través de los grupos con el problema inicialmente planteado.



Ilustración 1. Diagrama de Afinidad²

² http://upload.wikimedia.org/wikipedia/commons/7/75/Affinity_Diagram.png

Group sketching

El *group sketching* o dibujo en grupo, se presenta como una herramienta con la que poder aportar y explicar ideas de forma simultánea. Esta técnica se considera una herramienta de visualización de ideas caracterizada por su sencillez y rapidez a la hora de presentarlas [6]. Por otro lado, tiene un bajo coste económico debido al uso de pocos recursos y la nula necesidad de formación de las personas, ya que simplemente se trata de dibujar.

Los procesos comienzan con reuniones, generalmente en torno a una pizarra, donde los participantes pueden expresar y debatir sus ideas.

El objetivo de la utilización de esta técnica en las sesiones de codiseño es compartir puntos de vista de las diferentes personas que integran el grupo. Estos puntos de vista ofrecen una base común sobre la que trabajar fomentando la participación de cada uno de los miembros del equipo.



Ilustración 2. Group sketching³

Mind map

Un mapa mental es un diagrama utilizado para representar palabras, ideas o tareas dispuestas de forma radial alrededor de una palabra central. Su creación comienza con una idea en el centro

³ http://giffconstable.com/wp-content/uploads/jono_sketching.jpg

del mapa. A partir de esa idea o problema, se utilizan imágenes, palabras y líneas para construir un mapa mental.

El concepto fue creado por el psicólogo británico Tony Buzan y se considera como un método muy eficaz para extraer información. Se trata de una herramienta muy interesante para trabajar en sesiones colaborativas ya que permite contribuir al desarrollo de una misma idea o incluso encontrar el centro real de un problema a resolver.

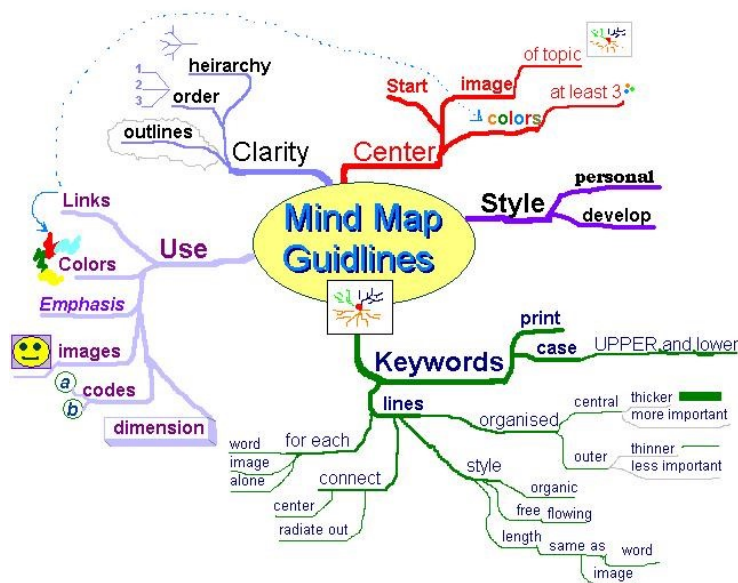


Ilustración 3. Mind map⁴

Motivation matrix

La matriz de motivación se presenta como una herramienta que facilita la investigación de una solución asumiendo el punto de vista y los intereses de cada uno de los *stakeholders* [6]. Para ello, cada uno de los actores expresa lo que espera o necesita del sistema. Este hecho permite comprender las conexiones entre cada uno de los actores interesados en el sistema.

El proceso de creación de la matriz de motivación comienza con el despliegue de una lista de motivaciones, beneficios y contribuciones de cada actor. La trazabilidad de estas listas permite obtener lo que cada actor puede ofrecer o requerir de la solución.

⁴ <http://upload.wikimedia.org/wikipedia/commons/2/26/MindMapGuidelines.JPG>

UNIVERSIDAD CARLOS III DE MADRID

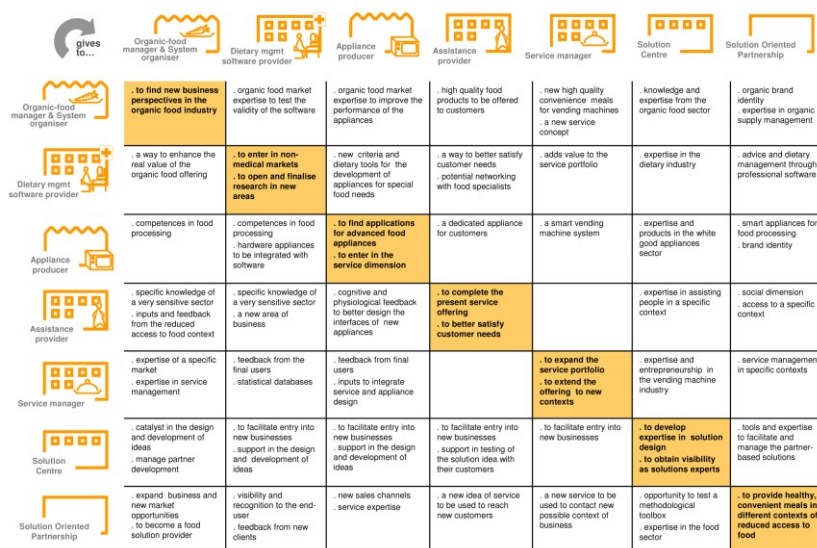


Ilustración 4. Motivation matrix⁵

Rough prototyping

El *rough prototyping* es un método para construir prototipos de forma rápida. Para su creación se utilizan objetos y materiales disponibles en el espacio donde se realiza la actividad. Los elementos creados se utilizan para simular los componentes de servicio con el fin de explicar mejor una idea frente al resto del equipo. Estos elementos son prototipos de baja fidelidad, como bocetos y maquetas que ayudan a contextualizar, idear y probar las ideas de diseño.

Se trata de una herramienta de apoyo a la visualización de ideas y una manera de estar seguro de que todos los miembros del equipo están hablando de lo mismo. A su vez, contribuye a que el proceso de diseño sea más interactivo y concreto.

⁵ http://www.servicedesigntools.org/sites/default/files/res_images/MOTIVATION_MATRIX_01.jpg



Ilustración 5. Rough prototyping⁶

Storyboard

El *storyboard* es un conjunto de ilustraciones mostradas en secuencia con el objetivo de servir de guía para entender y/o contar una secuencia de acciones o historia. Esta técnica proviene de la industria cinematográfica y fue desarrollada por los estudios Disney durante principios de los años 30 [6].

El proceso de creación colaborativo mediante el uso de *storyboard* cuenta con dos posibilidades. Una posibilidad es que los miembros del equipo acuerden un orden a partir del cual se van añadiendo nuevas viñetas basadas en las anteriores. La otra posibilidad se basa en que cada miembro crea su viñeta y luego se establece el orden de la secuencia de manera grupal.

Esta técnica es muy utilizada dentro del desarrollo del software como parte de la identificación de interfaces para un software particular. Para ello, se incluye dentro del equipo a los clientes. De

⁶ http://www.adobe.com/content/dotcom/en/devnet/fireworks/articles/cooper_prototyping/_jcr_content/articlecontentAdobe/image.img.jpg/1276068116872.jpg

este modo, se consiguen extraer nuevas características de la solución y permite que el cliente conozca el funcionamiento del producto antes de crearlo.

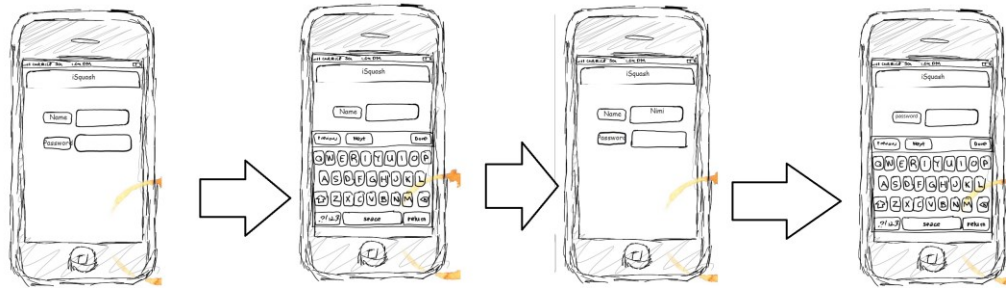


Ilustración 6. Storyboard⁷

Storytelling

El *storytelling* es una técnica empleada para contar historias desde un punto de vista narrativo en la que se busca conectar con la parte emocional del usuario. La técnica permite ilustrar la solución de un problema planteado y preparar los primeros bocetos para el *storyboard* [4].

Esta técnica es utilizada dentro del campo del desarrollo de software para establecer la comunicación de ideas. Así, a nivel colaborativo, mientras se cuenta la historia o el guion se dejan huecos en blanco para que sean rellenados por cada uno de los actores que cuentan con intereses en el sistema. De esta forma, los encargados van apuntando las características y necesidades que se van transmitiendo a medida que cada uno de los miembros del grupo va inventando o modificando la historia.

⁷ <http://spordipartner.files.wordpress.com/2011/10/storyboard-login.jpg>



Ilustración 7. Storytelling⁸

2.2 Revisión de arquitecturas móviles

En la actual “era de la información” todo el mundo necesita tener a mano la información que desee o sentirse en contacto con cualquier otra persona o empresa, viéndose esto reflejado en el mercado. En concreto, el índice de penetración de dispositivos móviles inteligentes en España fue del 55% en el primer trimestre de 2013 [7]. La previsión es que este índice continúe en aumento durante los próximos años. Entre las principales causas de este auge se encuentra el hecho de que los usuarios desean disponer de acceso a internet de forma permanente.

Debido a ello, es necesario que la industria del software ofrezca a los usuarios de estos nuevos dispositivos aplicaciones que cubran sus necesidades y ofrezcan una interacción adaptada a las características de este tipo de dispositivos. Por ello, en esta sección se va a llevar a cabo una revisión de las distintas arquitecturas móviles con el objetivo de determinar la mejor opción para desarrollar una aplicación móvil que permite asistir a los usuarios en los procesos de codiseño dentro del contexto del proyecto CoDICE.

⁸ http://www.servicedesigntools.org/sites/default/files/res_images/STORYTELLING.jpg

Revisión de arquitecturas para el desarrollo de aplicaciones móviles

Se va a comenzar la revisión de tecnologías móviles presentando las alternativas arquitectónicas a la hora de realizar la aplicación. Se van a estudiar los tres modelos arquitectónicos más utilizados: aplicaciones nativas, aplicaciones web y aplicaciones multidispositivo.

Aplicaciones nativas

Las aplicaciones nativas están diseñadas de forma específica para una plataforma (IOS, Android, Windows Phone, etcétera). Para ello, cada una de las plataformas ofrece a los desarrolladores un entorno de trabajo en el que se incluye un kit de desarrollo de software (SDK), un lenguaje propio y una tienda de aplicaciones.

	Android	IOS	Windows Phone
Kit de desarrollo	Android SDK	Xcode	Visual Studio
Lenguaje	Java	Objective-C, Swift	C#, VB .NET
Tienda	Google Play	Apple iTunes	Windows Phone Market

Tabla 1. Características aplicaciones nativas

Este modelo de desarrollo permite acceder a todas las características y elementos de los terminales (agenda de contactos, cámara, sensores, etcétera). Con ello se consiguen aplicaciones con un rendimiento superior al obtenido mediante el resto de alternativas de desarrollo.

Al mismo tiempo, es necesario destacar dos características muy importantes con las que cuentan las aplicaciones nativas. Por un lado, la facilidad de crear la aplicación con un diseño adaptado al sistema para el que se está desarrollando. Por otro, el hecho de contar con tiendas propias ya que facilita la distribución de las aplicaciones.

Aplicaciones web

Las aplicaciones web están orientadas a su ejecución desde cualquier dispositivo móvil mediante navegador web, por lo que la compatibilidad del sistema depende exclusivamente de las características del motor utilizado por el navegador. De hecho, no se trata de una aplicación como tal, ya que no es necesario instalarla en el dispositivo, siendo obligatorio contar con una conexión a internet para poder utilizar el sistema.

Para la creación de este tipo de aplicaciones se utiliza de forma mayoritaria el lenguaje de marcado HTML5, junto con CSS y JQuery Mobile (u otros *frameworks* similares basados en JavaScript) con los que se pretende presentar una vista cercana a la mostrada por las aplicaciones nativas.

Entre sus principales ventajas se encuentra la reducción en tiempo y coste para la creación y puesta en funcionamiento de la aplicación web en múltiples dispositivos, puesto que solo será necesario modificar pequeñas partes de código para crear una vista adaptada a los distintos dispositivos. Este sistema permite poder realizar actualizaciones sin que el usuario se vea afectado, aunque tiene como contrapunto el hecho de no poder encontrarse en las tiendas de aplicaciones de las distintas plataformas.

Aplicaciones híbridas

Este tipo de aplicaciones combina el uso de tecnologías nativas con tecnologías webs para la creación del producto.

Las aplicaciones web están escritas en HTML5, CSS y JavaScript. La aplicación web creada con estos lenguajes se integra en un contenedor que proporciona acceso a las características y hardware del móvil de un modo similar al que lo realiza una aplicación nativa.

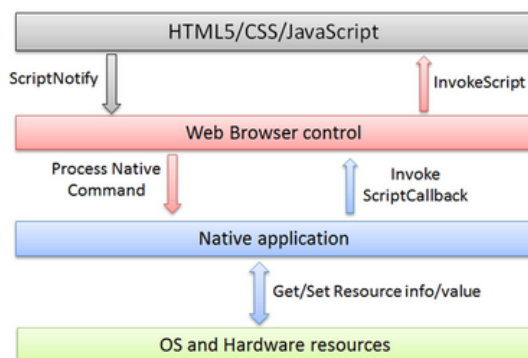


Ilustración 8. Arquitectura aplicaciones móviles híbridas⁹

⁹ http://windowsphone.interoperabilitybridges.com/media/140836/Windows-Live-Writer_Apache-Cordova-Introduction_A14A_clip_image001_44fec9ea-8359-4aa4-a767-776c0c8c975b.png

Los tiempos de desarrollo son menores que en el caso de las aplicaciones nativas, permitiendo crear aplicaciones multidispositivo con pequeñas variaciones en el código entre ellas. A diferencia de las aplicaciones web, estas aplicaciones pueden funcionar sin conexión a internet y ser publicadas en las tiendas de aplicaciones.

Evaluación de arquitecturas

Una vez presentadas las arquitecturas, se va a proceder a realizar una comparativa entre ellas. Para ello, en la [Tabla 2](#) se reflejan las principales diferencias entre estas:

	Aplicación nativa	Aplicación web	Aplicación híbrida
Gráficos	API nativa	HTML5	HTML5
Apariencia	Nativa	Emulada	Emulada
Distribución	Tienda	Web	Tienda
Actualizaciones	Requiere del usuario	Transparente a los usuario	Requiere del usuario
Acceso dispositivo	Completo, nativo	No	Sí, mediante APIs externas
Almacenamiento	En dispositivo	En caché y SQL compartido	En dispositivo y SQL compartido
Conectividad	Online y offline	Online	Online y offline
Conocimientos necesarios	Java, ObjectiveC, .NET	HTML5, CSS, JavaScript	HTML5, CSS, JavaScript

Tabla 2. Comparativa de arquitecturas

En la [Ilustración 9](#) se muestra de forma gráfica la situación a nivel de capacidades de cada una de estas tecnologías:

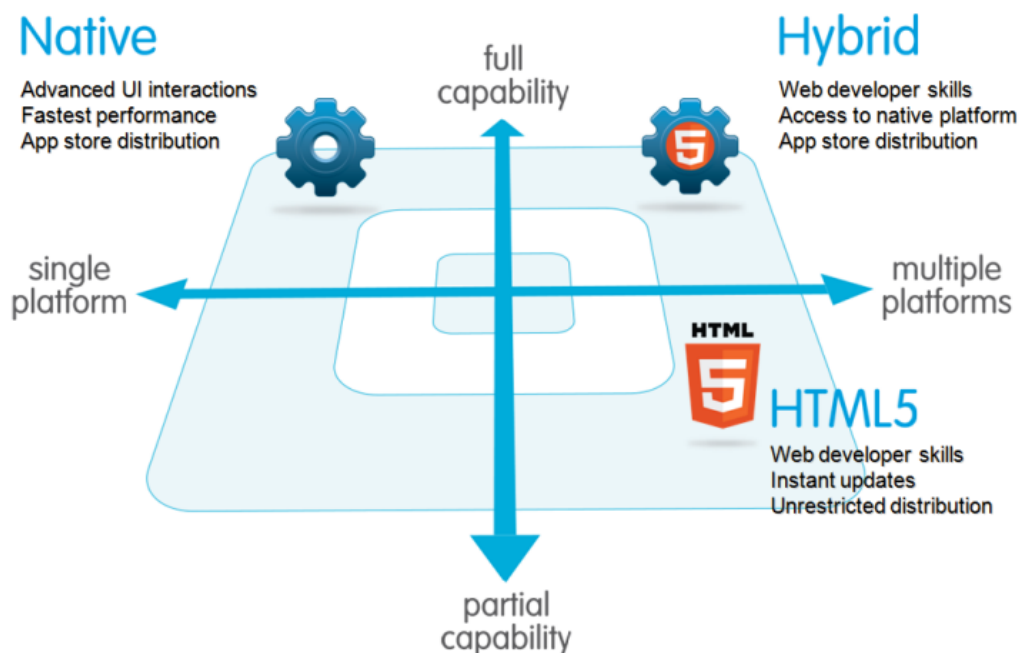


Ilustración 9. Situación arquitecturas móviles¹⁰

Tras observar las diferencias entre las alternativas se va a proceder a evaluar cada una de las propuestas. Para ello, para cada una de las alternativas se ha decidido evaluar si cumplen con los siguientes criterios extraídos de las necesidades del proyecto:

- **Experiencia previa.** Se considera importante contar con experiencia previa en el desarrollo con las arquitecturas estudiadas.
- **Alto rendimiento.** La alternativa elegida debe ofrecer un alto rendimiento para los usuarios en cualquier dispositivo que cumpla con los requisitos mínimos.
- **Multiplataforma.** Se valora el poder desarrollar la aplicación para varias plataformas de un modo rápido y eficaz.

¹⁰ <http://www.contunegocio.es/wp-content/uploads/2013/08/App-web-app-nativa-2.png>

	Experiencia previa	Alto rendimiento	Multiplataforma
Aplicaciones nativas	✓	✓	✗
Aplicaciones web	✓	✗	✓
Aplicaciones híbridas	✓	✓	✓

Tabla 3. Evaluación de arquitecturas

De acuerdo con la [Tabla 3](#) se puede observar que la arquitectura de aplicaciones híbridas es la única que satisface la totalidad de los criterios que se han estudiado. Por lo tanto, es la arquitectura elegida para desarrollar la aplicación móvil para el proyecto CoDICE.

2.3 Revisión de tecnologías móviles

Tras conocer la arquitectura sobre la que se va a trabajar es el momento de realizar la revisión de los distintos lenguajes existentes para la creación de aplicaciones móviles híbridas.

En esta sección se comienza con la descripción de HTML5 como lenguaje base sobre el que se desarrollan aplicaciones híbridas.

A continuación, se presenta una revisión del lenguaje JavaScript. Seguidamente, se comparan JQuery Mobile y AppFramework para la creación de interfaces de usuario. Por último, se continúa con la evaluación de dos nuevas librerías que utilizan JavaScript. En este caso, se van a evaluar las alternativas para la creación de un sistema distribuido en tiempo real mediante la utilización de *websockets*.

Lenguaje HTML5

El lenguaje HTML5 (*HyperText Markup Language*, revisión 5) es un estándar web utilizado ampliamente para el desarrollo de páginas web. Se trata de un lenguaje de hipertexto que utiliza

etiquetas o marcas a partir de las cuales se define el inicio y final de los elementos de la página para posteriormente poder ser interpretados de forma correcta por los navegadores web [8].

El enfoque del lenguaje a partir de su quinta revisión está orientado a simplificar el desarrollo y sobre todo, a incluir de forma nativa nuevos elementos multimedia. Este hecho permite que HTML5 no solo sea visto como un estándar web, sino como una de las grandes opciones a la hora de desarrollar aplicaciones para otros dispositivos como son dispositivos móviles, televisiones, etcétera.

JavaScript

JavaScript es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, débilmente tipado y dinámico. Este lenguaje se utiliza de forma principal dentro de una aplicación web, ofreciendo mejoras en las web dinámicas.

Existen un gran número de librerías basadas en JavaScript. Estas librerías aportan alguna funcionalidad extra de forma rápida por lo que resulta interesante valorar el uso según las necesidades de cada proyecto.

Revisión de alternativas para la creación de interfaces de usuario

A la hora de crear las interfaces de usuario de una aplicación móvil se presentan numerosas opciones a modo de librerías para JavaScript mediante las cuales se puede implementar una solución visual cercana a la ofrecida por las aplicaciones nativas. En esta revisión se van a estudiar y comparar dos de ellas, JQuery Mobile y AppFramework.

JQuery Mobile

JQuery Mobile es un sistema basado en JQuery que permite construir interfaces de usuario web siguiendo el patrón *responsive design* adaptado a los dispositivos móviles, ya que cuenta con optimizaciones para ser utilizado en superficies táctiles. Su diseño es muy cuidado y permite ofrecer a los usuarios una interfaz cercana a lo ofrecido por una aplicación nativa.

AppFramework

AppFramework ofrece una librería JavaScript orientada a la interfaz de usuarios para la creación de aplicaciones móviles híbridas multiplataforma. Cuenta con estilos de interfaces adaptados a los

principales ecosistemas móviles (Android, iOS, Windows Phone 8, etcétera) y permite crear muchos otros estilos mediante la compatibilidad que ofrece con los estándares de HTML5 y CSS.

Evaluación de librerías para creación de interfaces de usuario

En la [Tabla 4](#) se muestran las principales diferencias entre cada una de ellas:

	JQuery Mobile	AppFramework
Apariencia	Simula aplicación nativa	Simula aplicación nativa
Soporte oficial	jQuery.com	Intel
Comunidad de usuarios	Muy alta	Baja
Documentación	Excelente	Normal

Tabla 4. Comparativa librerías para la creación de interfaces

Una vez conocidas las diferencias, es necesario presentar las características sobre las que se van a evaluar cada una de las alternativas:

- **Alto rendimiento.** La alternativa elegida debe ofrecer un alto rendimiento para los usuarios en cualquier dispositivo que cumpla con los requisitos mínimos.
- **Experiencia de usuario satisfactoria.** Unido al rendimiento, es muy importante que la experiencia de usuario a la hora de navegar por la aplicación sea satisfactoria.

	Alto rendimiento	Experiencia de usuario satisfactoria
JQuery Mobile	✗	✗
App Framework	✓	✓

Tabla 5. Evaluación librerías para creación de interfaces de usuario

En la [Tabla 5](#) se puede ver como App Framework cumple con los dos requerimientos de este estudio. JQuery Mobile presenta carencias en cuestión de rendimiento y experiencia de usuario. En ambos criterios, JQuery Mobile solo ofrece el resultado deseado utilizando móviles de muy alta gama. Este hecho reduce de forma drástica la base de usuarios que pueden acceder a la aplicación con garantías de usabilidad y rendimiento adecuado.

Revisión de alternativas para la creación de un sistema distribuido en tiempo real mediante websockets

Una de las herramientas de comunicación que permiten realizar una conexión entre cliente y servidor con necesidades de muy baja latencia son los *websockets*. La especificación de la API de *websockets* ofrece una conexión persistente entre cliente y servidor pudiendo mandarse mensajes a un servidor de forma asíncrona y recibir respuestas controladas por eventos sin tener que consultar al servidor para obtener la respuesta.

A continuación se van a presentar las dos principales alternativas para incorporar *websockets* a un sistema cliente basado en JavaScript y un servidor creado con tecnologías .NET de Microsoft.

ASP.NET SignalR

SignalR es una biblioteca para desarrolladores de ASP.NET que facilita la implementación de sistemas en tiempo real permitiendo la comunicación bidireccional entre cliente y servidor compatible con *websockets*.

La API que proporciona SignalR está diseñada para la creación de llamadas a procedimiento remoto de servidor a cliente (RPC) mediante las que se llama a las funciones de JavaScript que se encuentra en los navegadores de los clientes desde el servidor creado en .NET.

Xsocket

Xsocket ofrece herramientas para crear aplicaciones en tiempo real en la plataforma .NET de Microsoft utilizando *websockets* y WebRTC.

Se trata de una plataforma totalmente personalizable que cuenta con arquitectura propia de *plugins* mediante la que se facilita desarrollar nuevas funcionalidades enfocadas a las necesidades de cada proyecto.

Evaluación de alternativas para la conexión mediante websockets

En la [Tabla 6](#) se presenta una comparativa de las principales características entre Xsockets y SignalR:

	Xsockets	SignalR
Publicar/ Suscribirse	✓	✓
Servicios de Windows	✓	✓
API cliente	✓	✓
SSL/TLS	✓	✓
Compatibilidad con Silverlight	✗	✓

Tabla 6. Comparativa tecnologías de conexión mediante *websockets*

En la comparativa se observa que Xsockets no es compatible con la tecnología Silverlight de Microsoft. Debido a las características del proyecto CoDICE, esta situación hace que sea necesario optar por la solución ofrecida por SignalR sin necesidad de continuar con el análisis.

2.4 Resumen del estado de la cuestión y elección de tecnologías revisadas

A lo largo del estado de la cuestión se ha presentado las principales técnicas de codiseño. Igualmente, se han presentado las tecnologías móviles que permiten encontrar la mejor opción para la creación de la aplicación móvil dentro del proyecto CoDICE. Algunas de esas tecnologías cuentan con diversas alternativas, por lo que han sido evaluadas de forma rigurosa hasta llegar a la solución tecnológica final que se presenta en la [Tabla 7](#).

Revisión realizada	Selección
Arquitectura	Híbrida
Lenguaje base	HTML5
Diseño y desarrollo de interfaz	AppFramework
Conexión mediante sockets	ASP.NET SignalR

Tabla 7. Resumen elección tecnologías

3 Gestión del proyecto software

En esta sección se va a definir la gestión de proyecto asociada al producto. Se comienza con la presentación del proyecto y el estudio de la solicitud. A continuación se realiza una estimación de recursos humanos. Basado en la estimación se presenta un presupuesto donde se incluyen todos los costes asociados al proyecto. De este modo se pretende ofrecer un presupuesto ajustado a las condiciones del proyecto teniendo en cuenta todas las variables, desde los costes por trabajador hasta la situación económica actual.

3.1 Alcance del proyecto

Con el alcance del proyecto se pretende definir de forma precisa los objetivos y condiciones o restricciones existentes a la hora de llevarlo a cabo. Del mismo modo, se incluye una estimación de las tareas y recursos para presentar un presupuesto completo al cliente.

Definición del proyecto

Este proyecto supone la creación de una aplicación móvil distribuida enmarcada dentro del proyecto CoDICE. Dicha aplicación permitirá a los usuarios añadir nuevos recursos tales como fotografías, videos o enlaces, acceder a los mismos y compartirlos con el resto de usuarios. Una de las principales características del sistema es que su funcionamiento será distribuido en tiempo real, por lo que cualquier actualización será recogida de forma inmediata por cualquier usuario que esté utilizando la versión web o móvil del producto. En concreto, en este proyecto se centra la solución en la aplicación móvil destinada a ese contexto. La aplicación web queda fuera del alcance de esta solución.

A pesar de realizar la aplicación con tecnologías web, contará con una interfaz adaptada al tipo de dispositivo en el que se ejecute, teniendo presente las normas de estilo recomendadas para cada uno de las plataformas en las que estará disponible la aplicación a la hora de realizar el diseño.

Dentro del alcance de este proyecto, se lanzarán las versiones para Android en la tienda de aplicaciones Google Play sin que tenga coste para los usuarios.

Estudio de la solicitud

De las necesidades propuestas por el cliente nace la necesidad de presentar las principales restricciones con las que se encuentra el desarrollo el proyecto:

- **Restricciones tecnológicas.** El sistema a crear debe ser totalmente compatible con la versión web del proyecto.
- **Restricciones económicas.** Los costes del sistema deben ser acordes a la oferta presentada.
Véase:
 - Apartado [Presupuesto](#) del presente documento.
- **Restricciones operativas.** El sistema deberá concluir y entregarse al cliente en la fecha acordada.
Véase:
 - Apartado [Presupuesto](#) del presente documento.
- **Restricciones legales.** Se garantiza la confidencialidad de datos de carácter personal del cliente, así como de los usuarios finales del sistema cumpliendo la Ley Orgánica 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal¹¹.

3.2 Plan de trabajo

En esta sección se va a presentar la planificación de tareas asociada al proyecto. Esta planificación permite especificar cada una de las tareas y organizarlas de modo que se ajusten al tiempo disponible para el desarrollo del producto.

En la [Tabla 8](#) se muestra la definición de tareas asociadas a la creación del producto con las fechas planificadas de comienzo y fin para cada una de ellas. Del mismo modo, en la [Tabla 9](#) se presenta la planificación para la documentación asociada al proyecto.

Tarea	Comienzo	Fin
Estudio previo	01/05/14	14/05/14
Definición de objetivos	01/05/14	05/05/14
Estado de la cuestión	02/05/14	12/05/14
Estudio de viabilidad del sistema	08/05/14	14/05/14
Iteración 1	15/05/14	12/06/14

¹¹ <https://www.boe.es/boe/dias/1999/12/14/pdfs/A43088-43099.pdf>

Análisis 1	15/05/14	21/05/14
Diseño 1	22/05/14	29/05/14
Implementación 1	29/05/14	11/06/14
Pruebas 1	11/06/14	12/06/14
Hito 1. Presentación de prototipo funcional al cliente	12/06/14	12/06/14
Iteración 2	13/06/14	11/07/14
Análisis 2	13/06/14	17/06/14
Diseño 2	17/06/14	22/06/14
Implementación 2	23/06/14	08/07/14
Pruebas 2	08/07/14	11/07/14
Hito 2. Presentación de prototipo funcional al cliente	11/07/14	11/07/14
Iteración 3	12/07/14	30/07/14
Análisis 3	12/07/14	14/07/14
Diseño 3	14/07/14	15/07/14
Implementación 3	16/07/14	22/07/14
Pruebas 3	23/07/14	30/07/14
Hito 3. Presentación del producto al cliente	30/07/14	30/07/14

Tabla 8. Tareas asociadas a la creación del producto

Tarea	Comienzo	Fin
Gestión del proyecto - Documentación	01/05/14	31/08/14
Investigación inicial	01/05/14	15/05/14
Introducción	19/05/14	30/05/14
Estado de la cuestión	05/06/14	25/06/14
Gestión del proyecto	02/07/14	14/07/14
Solución	21/07/14	15/08/14
Conclusiones	18/08/14	23/08/14
Revisión de documentación	25/08/14	31/08/14

Tabla 9. Tareas asociadas a la creación del documento

El seguimiento asociado a esta planificación se encuentra detallado en el [Anexo II. Seguimiento de Trabajo Fin de Grado](#).

3.3 Estimación de recursos

En la estimación de tareas y recursos se presentan los distintos perfiles de recursos humanos vinculados al proyecto, las tareas que van a desarrollar y la estimación de tiempo que van a dedicar a cada una de ellas. El equipo de trabajo estará formado por los siguientes roles:

- **Jefe de proyecto.** Será el encargado de dirigir el proyecto en todos sus ámbitos.
- **Analista.** Será el encargado de reunirse con el cliente y obtener los requisitos de usuario, definiéndolos y especificando tanto los requisitos como los casos de uso.
- **Diseñador.** Definirá la arquitectura y la interfaz de la aplicación, presentando los prototipos que luego deberán ser desarrollados.
- **Desarrollador de software.** Implementará la solución diseñada.
- **Responsable de pruebas.** Se encargará de realizar las pruebas y validarlas según los criterios acordados por el analista.

Debido al tamaño y características del proyecto, alguno de los roles serán desempeñados por la misma persona. Por ello, la configuración final del equipo queda de la siguiente forma:

- Habrá una persona ejerciendo el rol de jefe de proyecto con una dedicación de 2 horas diarias durante los meses que dure el mismo. Este puesto estará ocupado por un Ingeniero Informático.
- Una persona tomará los roles de analista y diseñador. Realizará sus funciones durante 6 horas al día durante el primer y último mes del proyecto. Para este puesto se requiere un Ingeniero Informático.
- Los roles de desarrollador de software y responsable de pruebas serán tomados por una única persona. Teniendo en cuenta el desempeño necesario para llevar a cabo el proyecto y a la situación económica actual, este perfil será desempeñado por un becario que trabajará en jornada de 8 horas durante los 3 meses finales del proyecto.

3.4 Presupuesto

Una vez realizada la estimación de recursos se va a proceder a presentar el presupuesto asociado al proyecto.

Costes asociados a Recursos humanos

Los salarios estimados para el proyecto se presentan en bruto y de forma mensual siguiendo la siguiente ecuación:

$$\text{Salario bruto mensual} = \frac{\text{Coste}}{\text{Hora}} * \frac{\text{Horas}}{\text{Día}} * \frac{\text{Días}}{\text{Mes}}$$

Rol	Coste/hora (€)	Horas/día	Días/mes	Bruto mensual (€)
Jefe de proyecto	26,70 €	2	22	1.174,80 €
Analista/ Diseñador	14,80 €	4	22	1.302,40 €
Desarrollador/ Responsable de pruebas	5,00 €	8	22	880,00 €
Total				3.357,20 €

Tabla 10. Costes asociados a RRHH

En la [Tabla 11](#) y [Tabla 12](#) se detallan las bases de cotización, así como los tipos de cotización para el año 2014 fijadas por el Ministerio de Empleo y Seguridad Social¹². Se debe destacar que sólo se han añadido los grupos de cotización en los que se encuentran los miembros del equipo asociado al proyecto.

Bases de cotización contingencias comunes 2014			
Grupo de cotización	Categorías profesionales	Bases mínimas (euros/mes)	Bases máximas (euros/mes)
1	Ingenieros y Licenciados. Personal de alta dirección no incluido en el artículo 1.3.c) del Estatuto de los Trabajadores	1.051,50 €	3.597,00 €
4	Ayudantes no Titulados	753,00 €	3.597,00 €

Tabla 11. Bases de cotización contingencias comunes 2014

Tipos de cotización 2014			
Concepto	Empresa	Trabajadores	Total
Contingencias comunes	23,60%	4,70%	28,30%
Horas Extraordinarias Fuerza Mayor	12,00%	2,00%	14,00%
Resto Horas Extraordinarias	23,60%	4,70%	28,30%

Tabla 12. Tipos de cotización 2014

¹² http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm

El hecho de contar con un becario para el proyecto, hace que sea necesario presentar las bases de cotización especiales con las que cuentan estos colectivos. Del mismo modo que en el caso anterior, los datos que se presentan en la [Tabla 13](#) han sido obtenidos del Ministerio de Empleo y Seguridad Social español¹³.

Tipos cotización por contratos de formación 2014			
Concepto	Empresa	Trabajadores	Total
Contingencias comunes	30,52 €	6,09 €	36,61 €

Tabla 13. Tipos cotización por contratos de formación 2014

En la [Tabla 14](#) se presentan las cuotas de seguridad social asociadas a los distintos roles que existen en el proyecto.

Rol	Sueldo bruto mensual (€)	Base mínima (euros/mes)	Base cotizada (euros/mes)	Tipo de cotización	Cuota de cotización (€)
Jefe de proyecto	1.174,80 €	1.051,50 €	1.174,80 €	23,60%	277,25 €
Analista/Diseñador	1.302,40 €	1.051,50 €	1.302,40 €	23,60%	307,37 €
Desarrollador/Responsable de pruebas	880,00 €	753,00 €	880,00 €	30,52 €	30,52 €

Tabla 14. Cuotas de cotización a la Seguridad Social

El coste total del personal asociado al proyecto se muestra en la [Tabla 15](#). Para el cálculo de costes asociado a cada rol se ha utilizado la siguiente ecuación:

$$\text{Coste final} = (\text{Sueldo bruto mensual} + \text{Cuota cotización}) * \text{Meses}$$

¹³ http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Basesytiposdecotiza36537/index.htm

Rol	Sueldo bruto mensual (€)	Cuota cotización (€)	Coste total/mes	Meses	Coste final (€)
Jefe de proyecto	1.174,80 €	277,25 €	1.452,05 €	4	5.808,21 €
Analista/Responsable de pruebas	1.302,40 €	307,37 €	1.609,77 €	2	3.219,53 €
Diseñador/Desarrollador	880,00 €	30,52 €	910,52 €	3	2.731,56 €
Total					11.759,30 €

Tabla 15. Coste de personal

Costes de hardware

En este apartado se presentan los recursos de tipo hardware que han sido necesarios para la realización del proyecto para a continuación realizar un análisis de costes.

El material hardware asociado al proyecto es el siguiente:

- **Ordenadores sobremesa.** Se ha provisto de 3 ordenadores *HP Pro Pavilion 3400 Series* dotados con procesador Intel® Core™ i3-2120 CPU y 4GB de memoria RAM. Se incluye teclado y ratón.
- **Ordenador portátil.** ha sido un portátil *MSI GE40* con un procesador Intel® Core™ i7-4702MQ y 8GB de memoria RAM.
- **Pantallas de ordenador.** Se ha optado por utilizar tres pantallas *ACER V196L* de 19 pulgadas.
- **Dispositivos móviles con sistema operativo Android.** Se ha optado por un móvil Google Nexus 4 y una tablet Samsung Galaxy Note 10.1.
- **Impresora láser.** Para imprimir la documentación se ha optado por una impresora láser monocromo con conexión en red *OKI B401DN*.

El cálculo de costes asociado a los productos presentados tiene en cuenta el periodo de amortización para cada uno de ellos, siendo computable al proyecto los gastos que se reflejan al aplicar la siguiente ecuación:

$$\text{Coste asociado al proyecto} = \frac{(\text{Precio total} * \text{Duración del proyecto})}{\text{Periodo de amortización}}$$

Los costes totales de hardware se ven reflejados en la [Tabla 16](#).

Producto	Coste Unitario (€)	Unidades	Coste total (€)	Periodo amortización (meses)	Duración proyecto (meses)	Coste asociado al proyecto (€)
Ordenador HP Pro Pavilion 3400 Series	499,00 €	3	1.497,00 €	48	4	124,75 €
Portátil MSI GE40	1.299,00 €	1	1.299,00 €	48		108,25 €
Monitor ACER V196L	102,00 €	4	408,00 €	24		68,00 €
Móvil Google Nexus 4	280,00 €	1	280,00 €	24		46,67 €
Tablet Samsung Galaxy Note 10.1	379,00 €	1	379,00 €	24		63,17 €
Impresora láser OKI B401	118,90 €	1	118,90 €	24		19,82 €
Total						430,65 €

Tabla 16. Costes de hardware

Costes de software

Del mismo modo que con el hardware, en este apartado se precede a realizar la estimación de costes asociados al software. A continuación se presentan los programas utilizados durante el proyecto:

- **Windows 8.1.** Se han utilizado 4 licencias de Windows 8.1. Este el tipo de licencia que viene asociada a los ordenadores cuando se adquieren por lo que el coste va incluido en el precio de los equipos.
- **Microsoft Office 365 Pequeña Empresa Premium.** Con esta versión de la suite de ofimática de Microsoft se adquieren las licencias de las versiones de escritorio así como de las

versiones online, pudiendo además realizar el uso compartido de los archivos. Se puede pagar de forma mensual por número de usuarios.

- **Microsoft Office Project Pro para Office 365.** Herramienta de gestión de proyectos utilizada por el Jefe de Proyecto para administrar de forma eficaz la totalidad del mismo.

A la hora de realizar la elección de productos, se ha optado por licencias cuyo contrato permite tener acceso a los productos por el tiempo que dure el proyecto. De este modo no ha sido necesario estimar un periodo de amortización.

Producto	Coste Mensual (€)	Unidades	Duración proyecto (meses)	Coste asociado al proyecto (€)
Microsoft Office 365	12,00 €	4	4	48,00 €
Microsoft Office Project Pro	26,10 €	4	4	104,40 €
Total				152,40 €

Tabla 17. Costes de software

Costes de material fungible

Para este proyecto se ha realizado la estimación de costes de material fungible basada en la experiencia en proyectos anteriores del Jefe de Proyecto. En la [Tabla 18](#) se muestran dichos costes.

Costes de material fungible			
Producto	Cantidad	Precio unitario (€)	Precio Total (€)
Papel DIN-A 4	1 cajas	13,00 €	13,00 €
Tóner impresora láser	1 pack	38,00 €	38,00 €
DVD	1 packs	12,72 €	12,72 €
Otro material de oficina	2 packs	10,00 €	20,00 €
TOTAL			83,72 €

Tabla 18. Costes material fungible

Costes indirectos

En los costes indirectos o de operación se incluyen los gastos propios del alquiler, luz, gas e internet, así como la contratación de un servicio de mantenimiento y limpieza de las instalaciones. En este proyecto los costes indirectos suponen un 10% del coste de recursos humanos:

Costes indirectos		
Gastos de RRHH	Porcentaje aplicado	Costes totales
11.759,30 €	10%	1.175,93 €

Tabla 19. Costes indirectos

Presupuesto por partidas

En la [Tabla 20](#) se presentan los costes asociados a cada una de las partidas presentadas.

Descripción	Coste total (€)
Recursos humanos	11.759,30 €
Costes hardware	430,65 €
Costes de software	152,40 €
Costes de material	83,72 €
Costes indirectos	1.175,93 €
Total	13.602,00 €

Tabla 20. Presupuesto por partidas

Costes asociados al riesgo

Al presupuesto diseccionado por partidas es necesario aplicarle una prima de riesgo sobre el total de sus gastos. En esta ocasión este valor está fijado en un 11% sobre el coste total del proyecto.

Costes asociado al riesgo		
Coste total (€)	Porcentaje aplicado	Costes asociados al riesgo (€)
13.602,00 €	11%	1.496,22 €

Tabla 21. Costes asociados al riesgo

Beneficio

El presupuesto cuenta con una partida asociada al porcentaje de beneficio. En este proyecto se aplica un porcentaje del beneficio del 15%, quedando por debajo de la media de proyectos de similares características. De este modo, se ofrece un presupuesto ajustado y adaptado a la situación económica actual.

Beneficio		
Coste total (€)	Porcentaje aplicado	Beneficio (€)
13.602,00 €	15%	2.040,30 €

Tabla 22. Beneficio

Resumen presupuesto

En la [Tabla 23](#) se presenta un desglose detallado del presupuesto asociado al proyecto incluyendo el precio final establecido a presentar como oferta económica al cliente.

PRESUPUESTO DETALLADO	
RECURSOS HUMANOS	11.759,30 €
RECURSOS FÍSICOS	666,77 €
COSTES INDIRECTOS	1.175,93 €
COSTES ASOCIADOS AL RIESGO	1.496,22 €
BENEFICIOS	2.040,30 €
TOTAL (ANTES IMPUESTOS)	17.138,53 €
IMPUESTOS (IVA 21%)	3.599,09 €
TOTAL	21.935,32 €

Tabla 23. Presupuesto detallado

Método de pago

Para facilitar el pago de los costes, se ha fraccionado el mismo en 3 cuotas. La primera de ellas será entregada a la firma del contrato. Se realizará un pago intermedio y el restante será ingresado una vez haya sido entregado el producto al cliente.

En la [Tabla 24](#) vienen reflejadas las fechas y cuotas de pago acordadas con el cliente.

Método de pago			
Cuotas	Porcentaje	Fecha de pago	Cantidad (IVA incluido)
Primera cuota	15%	01/05/2014	3.290,30 €
Segunda cuota	35%	22/07/2014	7.677,36 €
Tercera cuota	50%	31/08/2015	10.967,66 €

Tabla 24. Método de pago

El presupuesto para la realización de este proyecto asciende a la cantidad de **VEINTIUN MIL NOVECIENTOS TREINTA Y CINCO EUROS CON TREINTA Y DOS CENTIMOS (21.935,32 €) IVA incluido.**

4 Solución

En esta sección se detalla el proceso que se ha seguido para obtener la solución del sistema. La estructura de este apartado comienza con la descripción genérica de la solución donde se presentan los objetivos de la misma. A continuación, se expone el proceso de desarrollo de la solución. En dicho proceso se pueden diferenciar diferentes fases como el **análisis del sistema**, el **diseño** y la **implementación**. Estas fases serán explicadas y detalladas a lo largo de esta sección.

4.1 Descripción de la solución

El objetivo de la solución es asistir el proceso de codiseño a través de dispositivos móviles dentro del proyecto CoDICE.

La solución ofrece a los usuarios la posibilidad obtener y compartir nuevas ideas en forma de recursos multimedia desde sus dispositivos móviles. Los recursos pueden ser videos, fotos, documentos o direcciones URI capturadas de forma directa desde los dispositivos. Conjuntamente, la aplicación ofrece a los usuarios la capacidad de:

- **Compartir recursos.** Tras capturar un recurso con el dispositivo móvil este es privado. El sistema permite al usuario compartir este recurso con el resto de usuarios.
- **Etiquetar recursos.** El usuario puede crear y/o utilizar diversas etiquetas o *tags* para un recurso. El sistema de etiquetas pretende facilitar la ordenación y búsqueda de recursos según las *tags* a los que se encuentre asociado.
- **Modificar recursos.** El usuario puede modificar el nombre o descripción de los recursos accediendo a ellos.
- **Evaluar recursos.** Se presenta al usuario la posibilidad de valorar si un recurso le gusta o no mediante una serie de botones activos para ese cometido.
- **Comentar recursos.** Existe un sistema de comentarios que permite fomentar debate sobre los recursos publicados.
- **Actualizar datos personales.** En la aplicación, hay un módulo específico en el que el usuario puede modificar sus datos personales.

Todas estas características han sido creadas con soporte distribuido en tiempo real con el objetivo de favorecer el trabajo colaborativo para el que fue diseñado el sistema. Además, todo el sistema se encuentra sincronizado con la versión web de CoDICE.

4.2 El proceso de desarrollo

En esta sección se va a tratar el modelo de proceso elegido para el proyecto. A continuación, se va a desarrollar el análisis y diseño del sistema para finalizar con la implementación del mismo.

Modelo del proceso

El ciclo de vida del proyecto de software utilizado en este proyecto es un modelo iterativo e incremental. El modelo se basa en una serie de fases realimentadas y aplicadas de forma iterativa. De esta forma, se tiene en cuenta que el modelo se sigue por iteraciones. Estas iteraciones son pequeños desarrollos del modelo completo que se dan por finalizados con una versión parcial del producto con cierta funcionalidad ya implementada. Este modelo de ciclo de vida supone ventajas a la hora de reducir la tasa de fallos, mejorar la productividad del equipo y manejar la complejidad del proyecto ajustando cada iteración a las necesidades del proyecto.

La decisión de tomar este modelo de ciclo de vida para el proyecto viene dada por la necesidad de presentar al cliente versiones previas del producto. En concreto, se establecen con el cliente dos reuniones intermedias donde se presentan versiones parciales del producto y, a partir de las cuales, se negocian o modifican nuevos requisitos.



Ilustración 10. Ciclo de vida iterativo incremental¹⁴

¹⁴<http://3.bp.blogspot.com/ujJBNqRPxBc/UBgus4IRVnl/AAAAAAAAACo/aTEpOgeZMbE/s1600/Captura+de+pantalla+completa+31072012+141306.jpg>

Análisis

El análisis establece las necesidades y condiciones del sistema y se considera fundamental para la viabilidad del proyecto, ya que es la base sobre la que se realiza las etapas de diseño e implementación del producto. Esta fase lleva a la presentación de un conjunto de requisitos que permitan resolver el problema inicialmente planteado.

A lo largo de esta sección, se expone la definición de requisitos y se presentan los casos de uso asociados a cada uno de los mismos.

Definición de requisitos

En la definición de requisitos se presentan las funciones y restricciones del sistema. Para ello, los requisitos son agrupados en funcionales y no funcionales. Los primeros definen el comportamiento del sistema mientras que los segundos presentan condiciones o restricciones sobre los anteriores.

Para la especificación de requisitos, se ha utilizado una adaptación de la plantilla de Especificación de Requisitos de Volere [9] adaptada a las necesidades del sistema:

Identificador	RF-XX/RNF-XX		
Prioridad		Tipo de Requisito	
Descripción			
Prueba de verificación			

Tabla 25. Tabla de especificación de requisitos

En la [Tabla 25](#) se presentan los campos que se consideran necesarios para la especificación de los requisitos:

- **Identificador.** Muestra la identificación del requisito. Este atributo está compuesto por dos letras en caso de que se trate de un requisito funcional (RF) o tres letras si se trata de un requisito no funcional (RNF). Estas letras vendrán acompañadas por un guion y una serie de dos dígitos autoincrementales que permitirán la identificación del requisito.
 - **Requisito funcional:** RF(Requisito Funcional) – XX (digito1digito2)
 - **Requisito no funcional:** RNF(Requisito No Funcional) – XX (digito1digito2)

- **Prioridad.** Establece el grado de importancia del requisito en función del proyecto. Puede tomar los valores:
 - **Alta.** Imprescindible para conseguir los objetivos marcados del proyecto.
 - **Media.** Necesario para cumplir los objetivos del proyecto pero no primordial.
 - **Baja.** No influye de forma notable en la consecución de objetivos del proyecto.
- **Tipo de Requisito.** Define a quién afecta el requisito, ya sea el usuario o el sistema.
- **Descripción.** Describe el requisito dentro del contexto del proyecto
- **Prueba de verificación.** Describe la prueba que se ha de realizar para poder verificar cada uno de los requisitos.

A continuación se presenta en forma de lista la definición de requisitos que han surgido en cada una de las iteraciones del proyecto. La especificación detallada de requisitos detallada se adjunta en el [Anexo III. Especificación de requisitos](#).

Requisitos funcionales

A continuación se presenta la definición de los requisitos funcionales asociados al proyecto:

- **RF-01.** El sistema permitirá al usuario darse de alta.
- **RF-02.** El sistema permitirá al usuario acceder a la aplicación con su dirección de correo electrónico y contraseña.
- **RF-03.** El sistema permitirá al usuario recuperar su contraseña mediante su dirección de correo electrónico.
- **RF-04.** El sistema mostrará al usuario sus recursos privados.
- **RF-05.** El sistema mostrará al usuario los recursos compartidos por el resto de usuarios.
- **RF-06.** El sistema permitirá al usuario acceder a un recurso.
- **RF-07.** El sistema permitirá al usuario modificar el nombre del recurso.
- **RF-08.** El sistema permitirá al usuario modificar la descripción del recurso.
- **RF-09.** El sistema permitirá al usuario comentar recursos.
- **RF-10.** El sistema permitirá al usuario evaluar recursos.
- **RF-11.** El sistema permitirá al usuario asignar etiquetas.
- **RF-12.** El sistema permitirá al usuario crear nuevas etiquetas.
- **RF-13.** El sistema permitirá al usuario capturar y subir fotografías como nuevos recursos.

- **RF-14.** El sistema permitirá al usuario capturar y subir videos como nuevos recursos.
- **RF-15.** El sistema permitirá al usuario añadir y subir URIs como nuevos recursos.
- **RF-16.** El sistema permitirá al usuario añadir contenido multimedia alojado en el móvil como nuevos recursos.
- **RF-17.** El sistema permitirá al usuario modificar sus datos de acceso.
- **RF-18.** El sistema permitirá al usuario compartir los recursos privados mediante el botón compartir.
- **RF-19.** El sistema permitirá que las actualizaciones de los recursos se vean de forma inmediata por el resto de usuarios.

Requisitos no funcionales

De forma análoga a los requisitos funcionales, a continuación se presenta la definición de los requisitos no funcionales:

- **RNF-01.** El sistema será desarrollado para un tamaño de pantalla mínimo de 4 pulgadas.
- **RNF-02.** El sistema será creado con tecnologías web.
- **RNF-03.** Las actualizaciones de recursos deberán ser visibles por los usuarios en tiempo real.
- **RNF-04.** La interfaz será adaptada a los patrones de diseño de cada plataforma móvil.
- **RNF-05.** Las tecnologías del sistema deberán ser compatibles con ASP.NET y Microsoft Silverlight.

Casos de uso

En este apartado se presentan los casos de uso obtenidos a partir de los requisitos recogidos. Un caso de uso es la definición de los pasos a seguir por el usuario para realizar alguna tarea o proceso en la aplicación. En la [Tabla 26](#) se presenta la plantilla para la definición de los casos de uso:

Caso de uso	Identificador
Actores	
Escenario	
Descripción	
Precondiciones	
Pasos a seguir	

Tabla 26. Tabla de especificación de casos de uso

Los campos deben rellenarse según las siguientes especificaciones:

- **Caso de uso.** Presenta el nombre del caso uso.
- **Identificador.** Muestra la identificación única del caso de uso. Este atributo está compuesto por dos letras seguidas por un guion y una serie de dos dígitos autoincrementales:
 - CU-XX (Caso de Uso)-(Digito1Digito2)
- **Actores.** Define a los usuarios del sistema involucrados en el caso de uso.
- **Escenario.** Presenta el contexto en el que se produce el caso de uso.
- **Descripción.** Describe el caso de uso dentro del contexto especificado.
- **Precondiciones.** Define las condiciones previas necesarias para que comenzar el proceso.
- **Pasos a seguir.** Describe de forma detallada los pasos a seguir para llevar a cabo el proceso asociado al caso de uso.

A continuación se muestra la definición de casos de uso basada en los requisitos obtenidos en el apartado anterior de este documento:

- **CU-01.** El usuario accede a la aplicación desde su dispositivo móvil y se da de alta en el sistema.
- **CU-02.** El usuario accede al sistema.
- **CU-03.** El usuario introduce los datos que le permiten recuperar su contraseña.
- **CU-04.** El usuario accede a un recurso desde la lista de recursos.
- **CU-05.** El usuario modifica el nombre de un recurso ya creado.
- **CU-06.** El usuario modifica la descripción de un recurso ya creado.

- **CU-07.** El usuario realiza un comentario de un recurso ya creado.
- **CU-08.** El usuario valora de forma positiva un recurso.
- **CU-09.** El usuario valora de forma negativa un recurso.
- **CU-10.** El usuario asigna una etiqueta ya creada a un recurso.
- **CU-11.** El usuario crea una etiqueta ya la asigna a un recurso.
- **CU-12.** El usuario realiza una foto con el dispositivo y la añade a recursos.
- **CU-13.** El usuario realiza un video con el dispositivo y la añade a recursos.
- **CU-14.** El usuario añade una nueva URI a recursos.
- **CU-15.** El usuario añade un nuevo recurso ya almacenado en su dispositivo móvil.
- **CU-16.** El usuario añade información a un nuevo recurso subido al sistema.
- **CU-17.** El usuario comparte un recurso privado.
- **CU-18.** El usuario cambia alguno de sus datos de acceso.

La especificación detallada de casos de uso se encuentra adjunta en el [Anexo IV. Especificación de casos de uso.](#)

Matriz de trazabilidad

En la [Tabla 27](#) se presenta la matriz de trazabilidad entre los requisitos funcionales y los casos de uso.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09	CU-10	CU-11	CU-12	CU-13	CU-14	CU-15	CU-16	CU-17	CU-18
RF-01	x																	
RF-02		x																
RF-03			x															
RF-04		x																
RF-05		x																
RF-06				x														
RF-07					x													

RF-08						x											
RF-09							x										
RF-10								x	x								
RF-11										x							
RF-12											x						
RF-13												x				x	
RF-14													x			x	
RF-15														x		x	
RF-16															x	x	
RF-17																	x
RF-18																	x
RF-19						x	x	x	x	x	x	x					

Tabla 27. Matriz trazabilidad RF-CU

Diseño

A lo largo de esta sección se presenta el diseño del sistema. En esta fase se muestran los modelos y representaciones de identidades que permiten guiar el proceso de implementación del sistema de modo que se ajuste a la fase de análisis anteriormente realizada. Para poder describir y exponer dichos modelos, en primer lugar se establece el alcance del diseño y la implementación dentro del proyecto CoDICE. A continuación, se incluye la arquitectura del sistema y las tecnologías utilizadas en el desarrollo de la solución. Para finalizar, se establecen los diseños del producto detallado siguiendo las iteraciones realizadas en el modelo de proceso utilizado.

Contexto y alcance del sistema

El producto a desarrollar se encuentra integrado dentro del proyecto CoDICE. Dicho proyecto cuenta con una aplicación web implementada con tecnologías .NET de Microsoft. El objetivo de la aplicación móvil es ofrecer al usuario un sistema distribuido que permita acceder y añadir recursos a la web desde cualquier dispositivo móvil.

La arquitectura del sistema para la aplicación web queda reflejada en [Ilustración 11](#). En ella, se observa que el proyecto web se ha realizado mediante el encapsulado de Silverlight con controles ASP.NET. De este modo, se realiza la visualización de la web mediante Silverlight y el control del sistema queda en manos de ASP.NET que realiza la conexión con la base de datos mediante la herramienta de mapeo objeto-relacional NHibernate. Debido a las necesidades de crear un sistema distribuido en tiempo real se ha añadido la librería de *websockets* SignalR. Para finalizar, cabe destacar que toda la aplicación corre sobre el servidor de aplicaciones Internet Information Services (IIS).

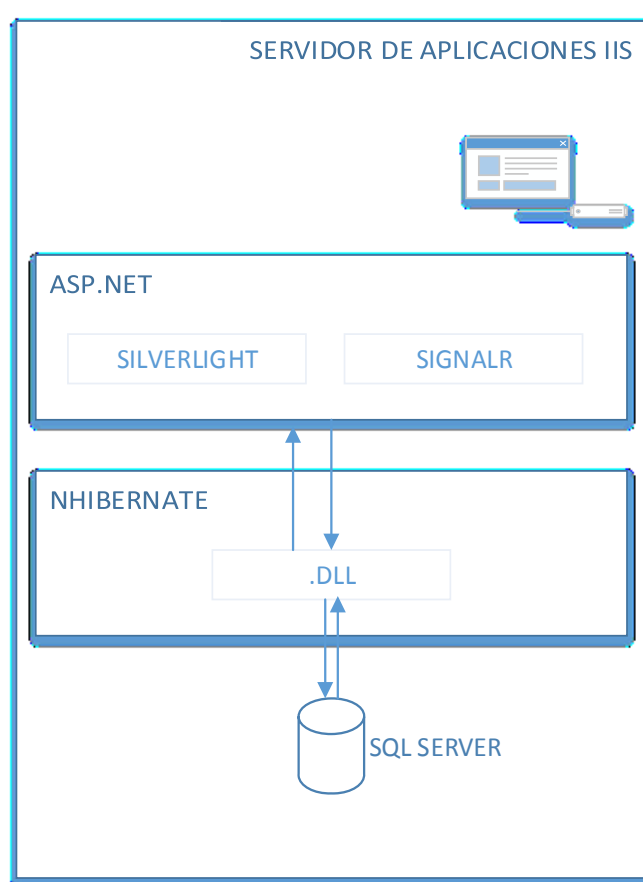


Ilustración 11. Arquitectura CoDICE

En la [Ilustración 12](#) se expone una aproximación a la arquitectura del sistema incluyendo la aplicación móvil. Mediante esta ilustración se presenta de forma clara el alcance del diseño y la implementación que se va a realizar. Para ello, se ha llevado a cabo una representación de cada una de las partes del sistema:

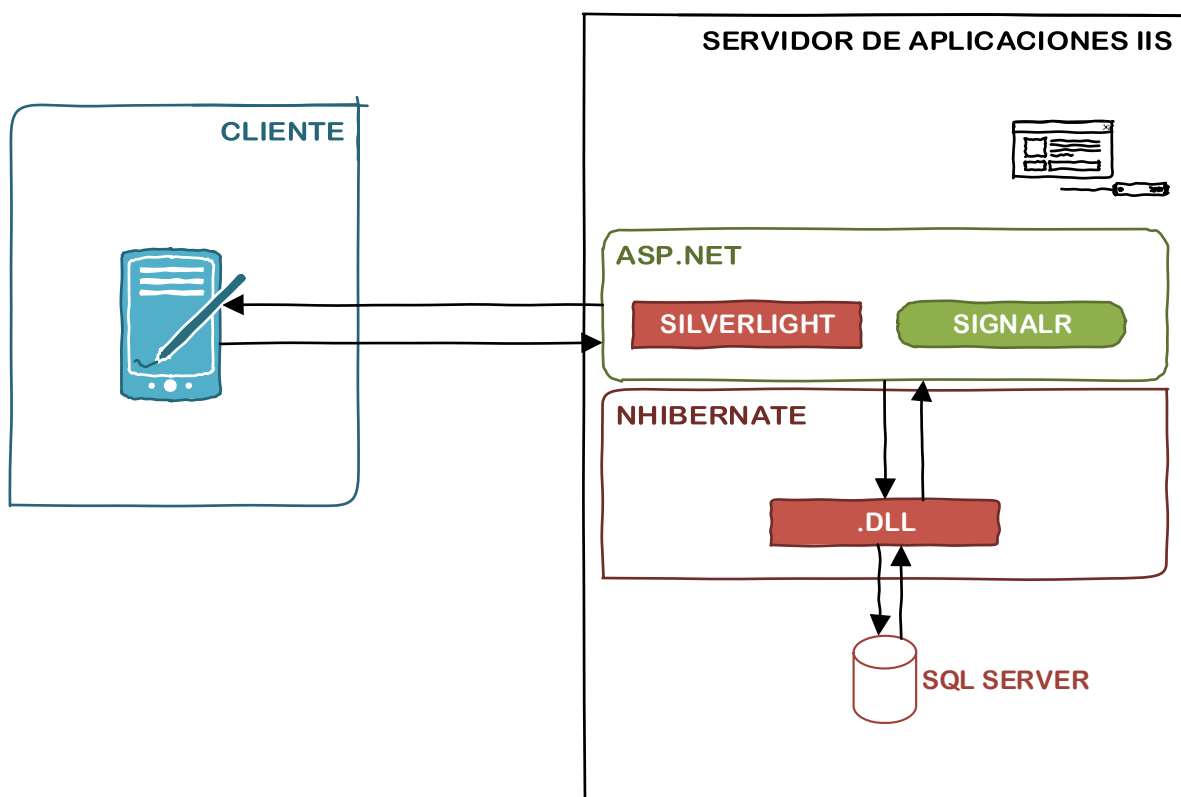


Ilustración 12. Alcance del sistema a implementar

El uso de colores limita el alcance del sistema. Cada una de las zonas representadas por los colores azules, verde y rojo se definen a continuación:

- **Zona azul.** Cubre el núcleo del producto que permite cubrir los objetivos marcados del proyecto. Se diseña e implementa en su totalidad.
- **Zona verde.** Esta zona no se desarrolla de forma directa ya que no se encuentra dentro del alcance del sistema. Aun así, es de vital importancia para la comunicación y la creación del sistema distribuido. Su definición, diseño e implementación se realiza de forma paralela con el equipo de trabajo encargado del proyecto web.
- **Zona roja.** No se encuentra dentro del alcance del sistema y los cambios no afectan al desarrollo del proyecto.

Es importante recalcar, sobre el contexto del sistema, que durante la primera reunión con el cliente se limitó el alcance del diseño e implementación del proyecto a las versiones Android para dispositivos móviles y tablets.

Diseño de sistema

En esta sección se presentan las tecnologías utilizadas y la arquitectura del sistema elegida para la realización de la solución.

Patrón de diseño - AngularJS

AngularJS es un *framework* de JavaScript orientado al desarrollo web *front-end* que permite construir aplicaciones siguiendo el patrón de diseño Modelo Vista Controlador (MVC) [10]. A su vez, ofrece la posibilidad de crear aplicaciones en una sola página HTML (*Single-Page Application*) y presenta alta compatibilidad con otras bibliotecas.

A pesar de contar con una curva de aprendizaje bastante elevada en comparación con otras bibliotecas, AngularJS ofrece características lo suficientemente atractivas para que su base de usuarios sea cada vez mayor.

El patrón Modelo Vista Controlador ofrecido por AngularJS define la organización independiente del Modelo (objetos de negocio), la Vista (interacción con el usuario u otro sistema) y el Controlador (controla el flujo de la aplicación) [11]. Con esta librería se separan las capas y el acceso a la parte servidor del sistema queda delegada al controlador.

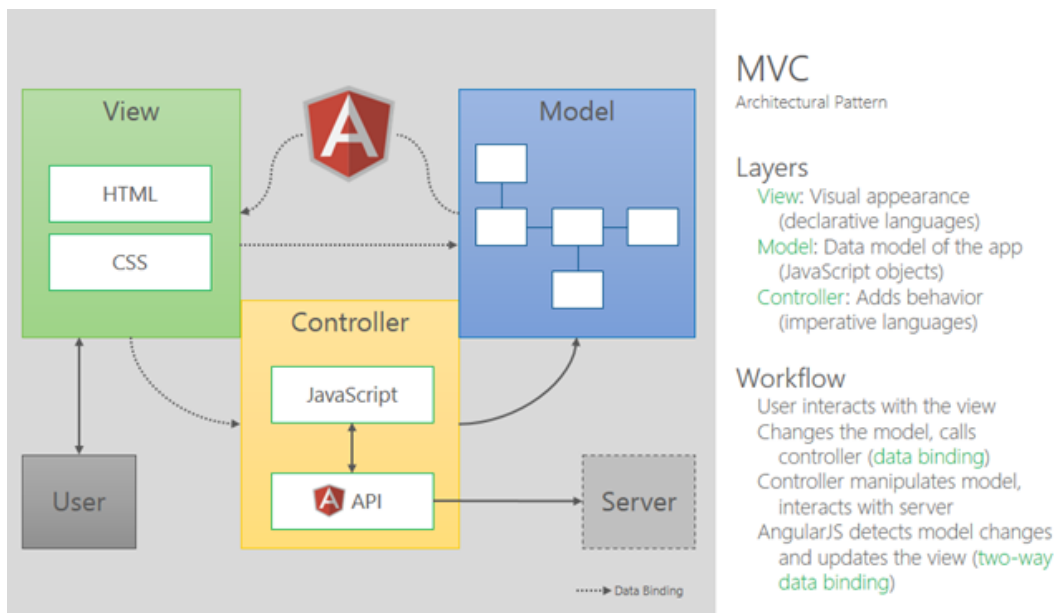


Ilustración 13. Modelo Vista Controlador AngularJS¹⁵

Controlador dispositivo móvil – Apache Cordova

Apache Cordova es un *framework* de código abierto para el desarrollo de aplicaciones móviles. Ofrece una API al desarrollador con la cual se accede a las funciones nativas de los dispositivos móviles mediante JavaScript. Gracias a esto se pueden construir aplicaciones móviles con tecnologías webs sin tener que recurrir al código nativo de cada dispositivo [12]. Entre las ventajas que ofrece al equipo de desarrollo, destacan las siguientes:

- Facilita la creación de aplicaciones para varios dispositivos sin tener que llevar a cabo un proyecto para cada plataforma.
- Permite empaquetar las aplicaciones y distribuirlas en las diferentes tiendas de aplicaciones.

La versión móvil de CoDICE necesita el acceso a diversas características nativas del dispositivo donde sea instalado. Por ello, la opción elegida para realizar estas tareas es la librería que ofrece Apache Cordova. Esta librería permite realizar fotos, grabar videos, obtener recursos ya disponibles

¹⁵ http://www.software-architects.com/media/59691225-2c60-4d19-9121-802712882d29/1837367778/time_cockpit/blog/2013/10/Slide7.PNG?mw=650&mh=650

en el dispositivo móvil y facilita la subida de recursos al servidor mediante varias de sus funciones disponibles.

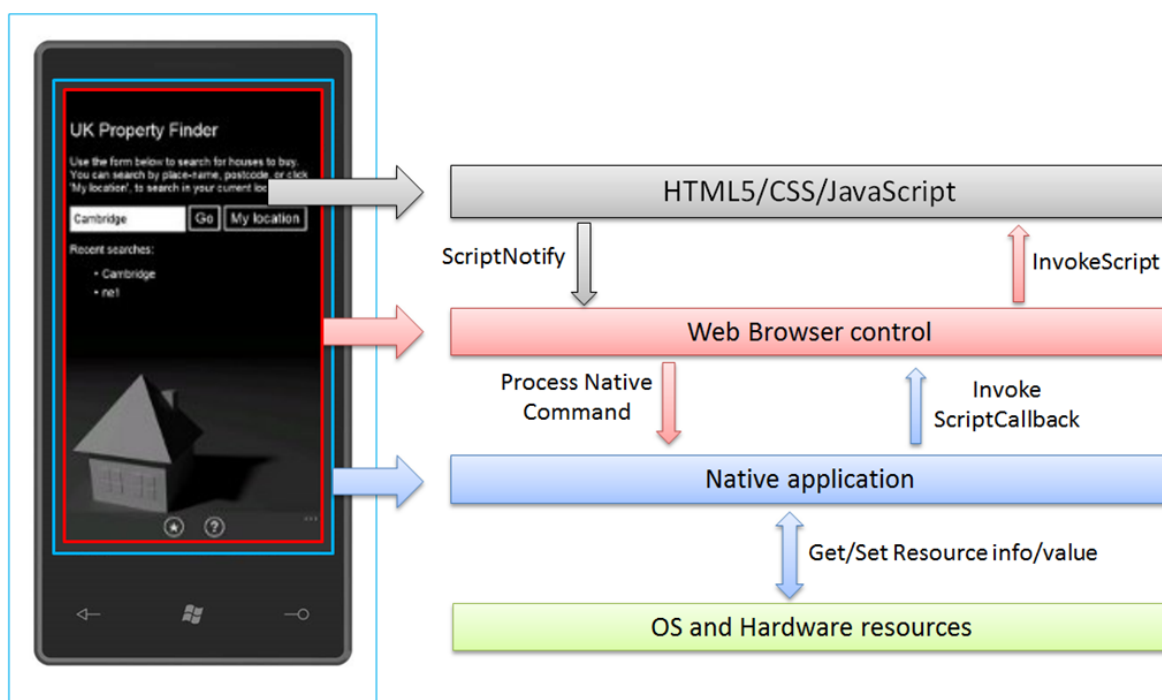


Ilustración 14. Encapsulado Apache Cordova

Comunicación

Para conseguir un sistema distribuido la arquitectura de comunicación es fundamental. En este proyecto se trabaja con el envío de mensajes mediante el uso de *websockets* y con la subida de recursos multimedia al servidor.

En el envío de mensajes mediante *websockets* se utiliza la librería SignalR. Esta tecnología permite realizar llamadas a procedimiento remoto desde un servidor a los clientes conectados y desde los clientes al servidor [13]. En el código de servidor se definen los métodos que pueden ser llamados por los clientes y se llama a los métodos que se ejecutan en el cliente. Por su parte, en el cliente se definen los métodos que se pueden llamar desde el servidor y se llama a los métodos que se ejecutan en el servidor. En definitiva, el uso de *Hubs* permite una conexión entre cliente y servidor para llamar métodos entre sí de forma directa.

En la [Ilustración 15](#) se muestra un diagrama de funcionamiento de SignalR mediante *Hubs* diseñado para el proyecto.

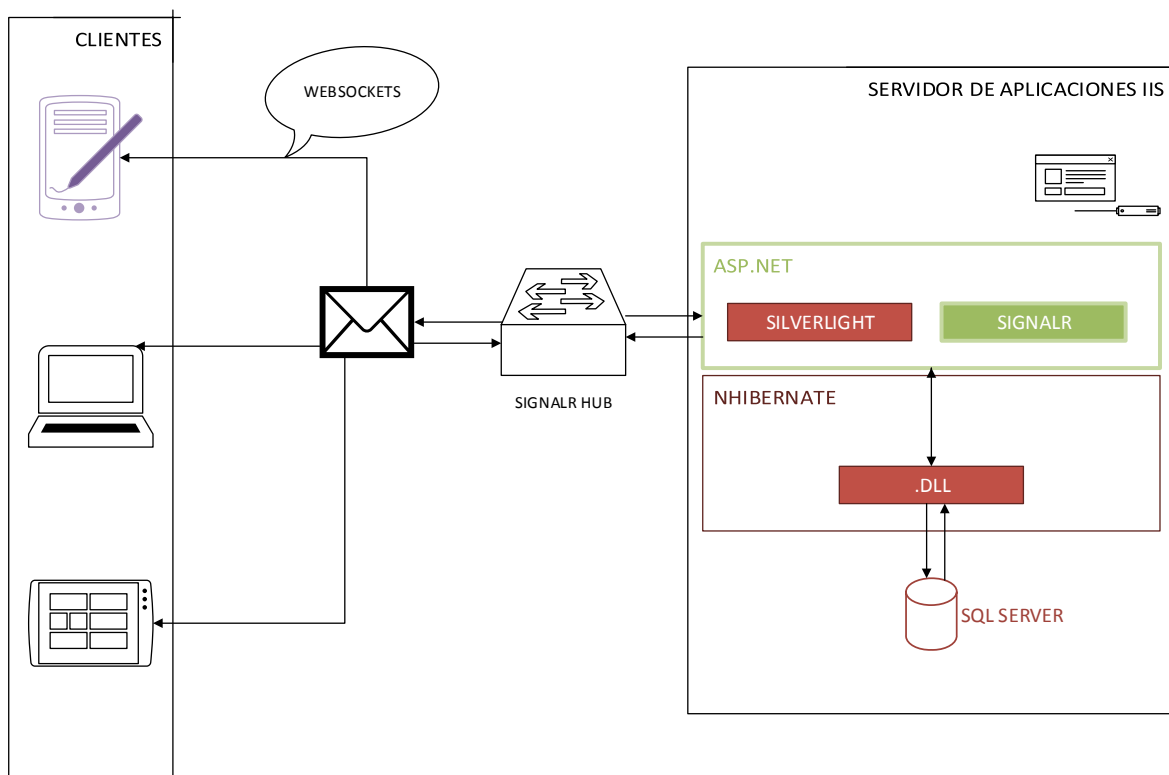


Ilustración 15. HUB SignalR

Por otro lado, la aplicación requiere de un sistema que permita subir recursos a la aplicación y por ende, también al servidor. Existen multitud de alternativas para realizar esta tarea pero la opción diseñada aprovecha las posibilidades ofrecidas por la API de Apache Cordova.

Esta librería proporciona un método para subir archivos a un servidor mediante una solicitud POST a través de la función FileTransfer [14]. En la [Ilustración 16](#) se muestra un ejemplo de petición POST sobre la capa ASP.NET. El tratamiento del recurso por parte del servidor lo lleva a cabo el controlador del servidor, el cual envía la información tanto a la vista de la aplicación web como a la base de datos mediante NHibernate.

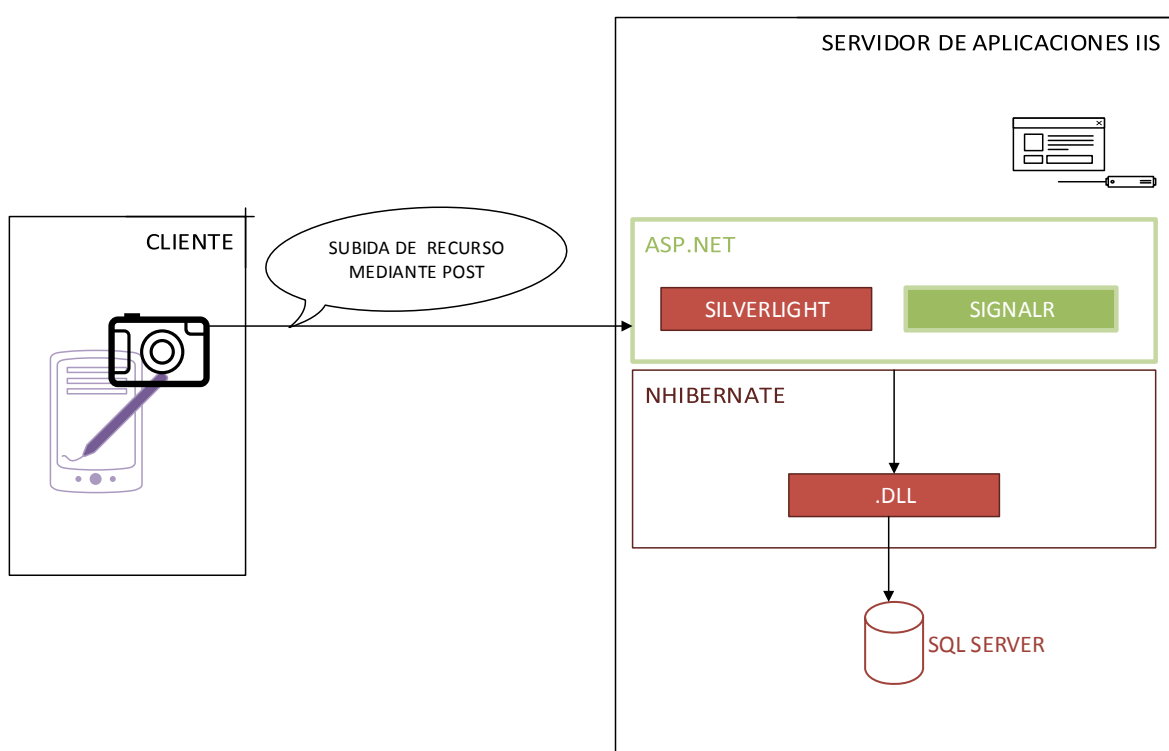


Ilustración 16. Subida de recursos mediante POST

Arquitectura del sistema

La arquitectura de una aplicación móvil realizada con tecnologías web se puede presentar como una arquitectura con diferentes capas que comienza con la presentación de la interfaz al usuario y finaliza con el control del hardware del dispositivo gracias a la librería Apache Cordova. En la [Ilustración 17](#) se observa un esquema de la arquitectura utilizada:

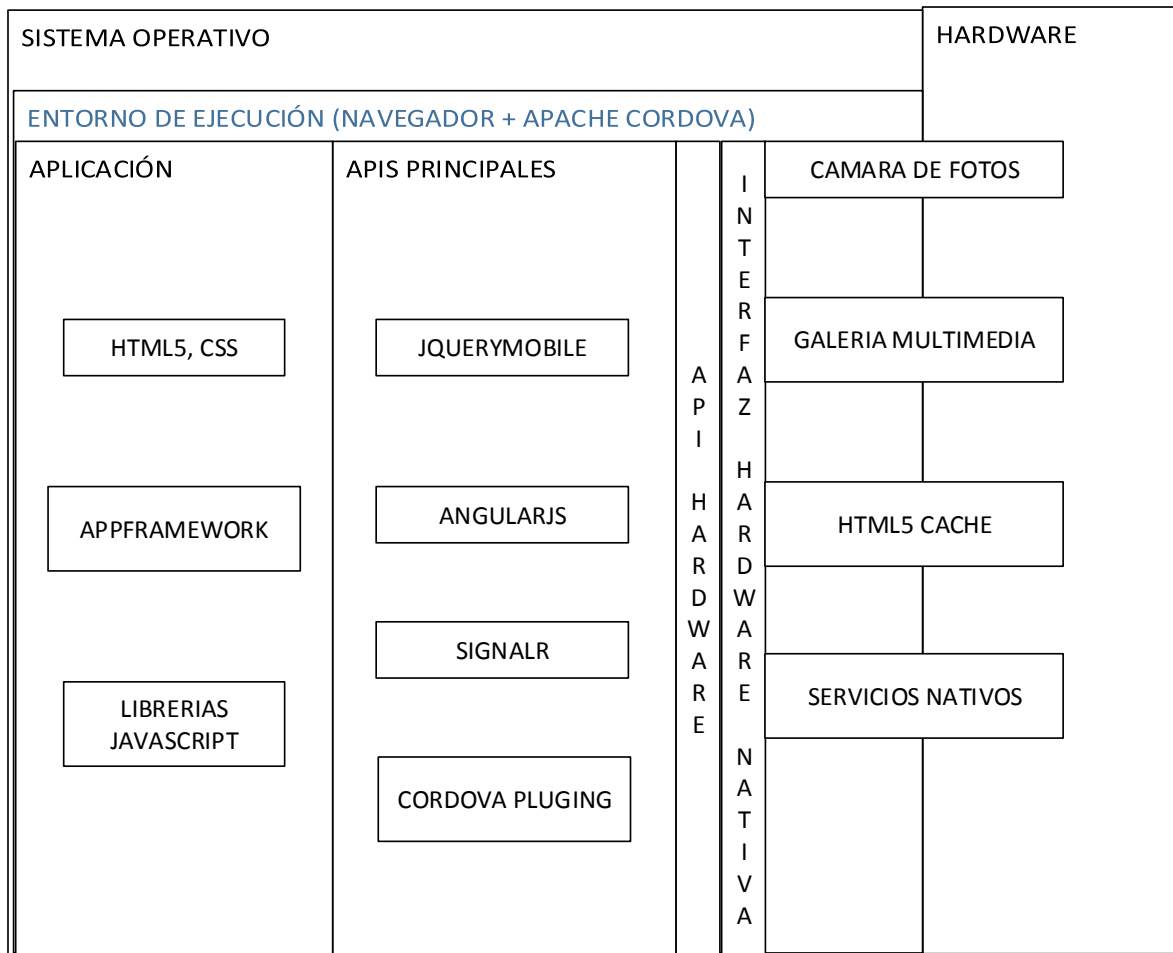


Ilustración 17. Arquitectura del sistema

En la **primera de las capas** se encuentran los archivos HTML y CSS que proporcionan la Vista del sistema. Para este proyecto se utiliza la librería de Intel® AppFramework para proporcionar una interfaz adaptada a dispositivos móviles con un diseño web adaptable que permita la correcta visualización del contenido desde cualquier dispositivo móvil [15]. Por último, en esta primera capa, se cargan las librerías de JavaScript que son utilizadas en la siguiente capa.

En la **segunda capa** de la arquitectura se ejecutan los *frameworks* basados en JavaScript. Estos *frameworks* proporcionan la conexión entre capas y el controlador de la aplicación. A continuación se expone cada uno de ellos:

- **JQuery Mobile.** El resto de librerías basadas en JavaScript necesitan de JQuery Mobile para su correcto funcionamiento.
- **AngularJS.** Proporciona a la aplicación el patrón Modelo Vista Controlador.

- **SignalR.** Permite crear un ecosistema distribuido en tiempo real mediante el uso de *websockets*.
- **Plugin Apache Cordova.** Proporciona una API que permite tener acceso a los controles nativos del dispositivo.

La **tercera capa** de la aplicación es la de hardware. A ella se tiene acceso mediante el uso de la API ofrecida por Apache Cordova. Esto es debido a que crea una interfaz de conexión hacia el dispositivo. Con esta interfaz se consigue acceso a la cámara de fotos, a la galería multimedia, a la caché generada por HTML5 y a otros servicios nativos del dispositivo.

Todas estas capas se encuentran envueltas en un entorno de ejecución formado por el navegador web y Apache Cordova. Este entorno se genera de forma automática cuando se realiza la construcción de la aplicación, obteniendo como resultado el fichero instalable para los dispositivos móviles.

Diseño detallado

En esta sección se presenta con mayor profundidad el diseño de la aplicación. En primer lugar se explica la exclusión en este diseño de la capa de persistencia. A continuación, se presenta una descripción del diseño de clases del sistema. Por último, se muestra la evolución en el diseño de la interfaz en cada una de las iteraciones sobre la solución.

Capa de persistencia

La capa de persistencia se encuentra fuera del alcance de este proyecto, como ya se ha indicado en [Contexto y alcance del sistema](#). El diseño e implementación de la aplicación web supone que el uso de la capa de persistencia ya este creada, por lo que, a la hora de llevar a cabo el proyecto de aplicación móvil, este se tiene que adaptar a los modelos ya creados. Para llevar a cabo este proceso se trabaja de forma paralela con el equipo de desarrollo encargado del sistema web y del servidor de la aplicación.

Diseño de la Vista

A la hora de diseñar las clases HTML se ha optado por el uso del paradigma de desarrollo Single Page Apps mediante el uso de las librerías AppFramework y AngularJS. El objetivo de este paradigma es mejorar la experiencia de usuario reduciendo los tiempo de espera o latencias entre cada una de

las vistas. Debido a esta elección, en la aplicación solo existe una clase HTML (index.html) en la que se recogen todas las vistas de la aplicación mediante el uso de paneles. La descripción de cada panel se presenta en una plantilla como la que se muestra en la [Tabla 28](#).

Panel	Controlador asociado
Descripción	
Componentes de diseño	

Tabla 28. Plantilla diseño vista

La plantilla para la definición de diseño de la vista se completa teniendo en cuenta la definición de cada uno de sus atributos:

- **Panel.** Nombre unívoco asociado a cada uno de los paneles presentes en el fichero HTML de la solución.
- **Controlador asociado.** Cada panel cuenta con un controlador asociado que permite seguir el patrón MVC proporcionado por la librería AngularJS.
- **Descripción.** Descripción de la vista a la que corresponde el panel.
- **Componentes de la vista.** Descripción de los componentes HTML con los que cuenta la vista.

Una vez presentada la plantilla, se detallan los paneles en las siguientes tablas:

Panel	Login	Controlador asociado	LoginController
Descripción	Se presenta la interfaz de acceso a la aplicación. El usuario puede introducir sus datos de acceso, ir a los paneles destinados a crear una nueva cuenta o recuperar la contraseña de una cuenta ya creada		
Componentes de la vista	<ul style="list-style-type: none"> - Imagen con el logotipo de CoDICE - Entrada de texto tipo email - Entrada de texto tipo contraseña - Botón de login - Texto en párrafo para solicitar vista de alta de usuario - Texto en párrafo para solicitar vista de recuperación de contraseña 		

Tabla 29. Diseño vista login

Panel	Sign-up	Controlador asociado	SignupController
Descripción	Se presenta la interfaz de alta de usuario. El usuario debe rellenar los campos disponibles para darse de alta en la aplicación		
Componentes de la vista	<ul style="list-style-type: none"> - Imagen con el logotipo de CoDICE - Entrada de tipo texto para el nombre - Entrada de tipo texto para el apellido - Dos entradas de texto tipo email - Dos entradas de texto tipo contraseña - Botón de alta de usuario 		

Tabla 30. Diseño vista sign-up

Panel	Password recovery	Controlador asociado	ForgotPasswordController
Descripción	Se presenta la interfaz que permite al usuario recuperar la contraseña. Debe rellenar el campo solicitado para obtener su nueva contraseña		
Componentes de la vista	<ul style="list-style-type: none"> - Imagen con el logotipo de CoDICE - Entrada de texto tipo email - Botón de recuperación de contraseña 		

Tabla 31. Diseño vista password recovery

Panel	Resources	Controlador asociado	ResourcesController
Descripción	Se presenta la vista con todos los recursos disponibles en formato lista. Se muestra a la izquierda una imagen previa del recurso y a la derecha el nombre del recurso y una breve descripción del mismo		
Componentes de la vista	<ul style="list-style-type: none"> - Lista de recursos con imagen previa a la izquierda de cada recurso - Botón superior izquierdo para abrir menú lateral - Botón superior derecho para agregar nuevos recursos 		

UNIVERSIDAD CARLOS III DE MADRID

Tabla 32. Diseño vista resources

Panel	Resource	Controlador asociado	ResourceController
Descripción	Se presenta toda la información asociada a un recurso. Permite al usuario realizar cambios en el nombre y en la descripción. Permite asociar etiquetas, evaluar el recurso y añadir comentarios		
Componentes de la vista	<ul style="list-style-type: none"> - Entrada de texto con el nombre del recurso con botón a la derecha para compartir recurso - Imagen del recurso - Barra de progreso con contador de <i>likes</i> y <i>dislikes</i> del recurso - Botón de <i>like</i> - Botón de <i>dislike</i> - Entrada de texto para <i>tags</i> con botón a la derecha para añadir nueva etiqueta - Campo de texto para añadir descripción del recurso - Campo de texto para añadir comentario con botón a la derecha para confirmar el comentario - Lista de comentarios añadidos al recurso - Botón superior izquierdo para regresar a la lista de recursos 		

Tabla 33. Diseño vista resource

Panel	Add Resource	Controlador asociado	AddResourcesController
Descripción	Se presenta la interfaz con un menú que incluye el tipo de nuevos recursos que se puede añadir a la aplicación		
Componentes de la vista	<ul style="list-style-type: none"> - Lista con los diversos tipos de nuevo recursos que se pueden añadir (fotografía, video, archivo del dispositivo y URI) - Botón superior izquierdo para regresar a la lista de recursos 		

Tabla 34. Diseño vista add resource

UNIVERSIDAD CARLOS III DE MADRID

Panel	New URI	Controlador asociado	AddResourcesController
Descripción	Se presenta la interfaz para añadir un recurso de tipo URI. Debido a que el recurso se añade en formato texto se ha diseñado un nuevo panel aunque el controlador es el mismo que para el panel Add Resource		
Componentes de la vista	<ul style="list-style-type: none"> - Campo de texto con botón a la derecha para confirmar la introducción de la nueva URI - Botón superior izquierdo para regresar a la lista de añadir recursos 		

Tabla 35. Diseño vista new URI

Panel	Account	Controlador asociado	AccountController
Descripción	Se presenta la vista con la información del usuario. Esta información la puede cambiar en cualquier momento introduciendo la contraseña actual		
Componentes de la vista	<ul style="list-style-type: none"> - Entrada de tipo texto para el nombre - Entrada de tipo texto para el apellido - Entrada de texto tipo email - Tres entradas de texto tipo contraseña (dos para contraseña actual y una para nueva contraseña en caso de que se desee cambiar) - Botón de alta de usuario Botón superior izquierdo para regresar a la lista de añadir recursos 		

Tabla 36. Diseño vista account

Panel	Left navigation	Controlador asociado	SlideController
Descripción	Se presenta la vista del menú desplegable que se encuentra oculto en la zona izquierda de la aplicación. Permite tener acceso directo a la cuenta de usuario y a los recursos. Permite salir de la aplicación.		
Componentes de la vista	<ul style="list-style-type: none"> - Menú tipo lista con acceso a la cuenta de usuario, a la lista de recursos y salida de la aplicación mediante <i>log-out</i> 		

Tabla 37. Diseño vista left navigation

Diseño del Controlador

En esta sección se presenta el diseño del controlador del dispositivo. Tal y como se ha definido en el apartado [Diseño de sistema](#) este producto utiliza las librerías AngularJS para ofrecer una patrón MVC, SignalR para obtener el sistema distribuido y Apache Cordova para el control de las características internas del dispositivo.

Todas estas librerías son utilizadas a la hora de diseñar el controlador del dispositivo. A su vez, para conseguir la conexión con el sistema web se ha adaptado el diseño a la funcionalidad mediante el *Hub* proporcionado por SignalR. Toda la funcionalidad que se va a mostrar se encuentra dentro de un fichero de JavaScript denominado angularController.js

En la [Tabla 38](#) se presenta la plantilla con la que se va presentar la funcionalidad del controlador del sistema.

Controlador		
Atributos		
Funciones		

Tabla 38. Plantilla diseño controlador

La plantilla para la definición de diseño del controlador se rellena según la siguiente definición de los atributos:

- **Controlador.** Nombre unívoco asociado a los controladores diseñados para el fichero JavaScript.

- **Atributos.** Se muestran los atributos utilizados por el sistema. AngularJS ofrece la correlación entre la vista, el controlador y el servidor de la aplicación utilizando la variables \$scope.
- **Funciones.** Describe el diseño de la funcionalidad para el sistema.

A continuación, se muestra la descripción detallada de cada controlador:

Controlador	LoginController	
Atributos	\$scope.user	Usuario del sistema identificado mediante un identificador
Funciones	login	Mediante esta función se realiza la entrada en el sistema cargando la lista de recursos disponibles

Tabla 39. Diseño controlador LoginController

Controlador	SignUpController	
Atributos	\$scope.user	Usuario del sistema identificado mediante un identificador
Funciones	signUp	Esta función permite crear una nueva cuenta de usuario

Tabla 40. Diseño controlador SignUpController

Controlador	ForgotPasswordController	
Atributos	\$scope.email	Correo electrónico asociado al usuario
Funciones	forgotPassword	Esta función permite recuperar la cuenta de usuario introduciendo el correo electrónico

Tabla 41. Diseño controlador ForgotPasswordController

Controlador	ResourcesController	
Atributos	\$scope.resources	Array de recursos disponibles en el sistema
	\$scope.loaded	Booleano que indica si ha sido cargado
Funciones	getAllResourceItems	Esta función obtiene la lista de recursos del sistema
	subscribeResource	Esta función suscribe los recursos a los usuarios

Tabla 42. Diseño controlador ResourcesController

Controlador	ResourceController	
Atributos	\$scope.resource	Recurso del sistema identificado mediante una id
	\$scope.user	Usuario del sistema identificado mediante una id
	\$scope.newComment	String de tipo comentario asociado a un recurso
Funciones	getResourceItemById	Esta función obtiene un recurso por su identificador
	getAllTag	Esta función obtiene todas las etiquetas del sistema
	share	Esta función permite compartir un recurso privado y lo convierte en un recurso compartido

	change	Esta función permite controlar los cambios que se realizan en los distintos campos de un recurso
	deleteTag	Esta función permite eliminar una etiqueta ya creada
	selectNewTag	Esta función permite selección una etiqueta ya creada
	addNewTag	Esta función permite añadir una nueva etiqueta al sistema
	ResourceLike	Esta función permite controlar la barra de progreso de valoración de un recurso con <i>like</i> o <i>dislike</i>
	InsertResourceComment	Esta función permite añadir un comentario a un recurso

Tabla 43. Diseño controlador ResourceController

Controlador	AddResourcesController	
Atributos	\$scope.email	Correo electrónico asociado al usuario
Funciones	capturePhoto	Esta función utiliza las características internas del dispositivo móvil para realizar una foto
	captureVideo	Esta función utiliza las características internas del dispositivo móvil para realizar un video

	getResource	Esta función utiliza las características internas del dispositivo móvil para obtener el recurso tras haber sido realizada la foto o el video
	uploadFile	Esta función sube los recursos

Tabla 44. Diseño controlador AddResourcesController

Controlador	AccountController	
Atributos	\$scope.user	Usuario del sistema identificado mediante un identificador
Funciones	currentUser	Esta función obtiene la información del usuario
	updateUser	Esta función actualiza la información del usuario

Tabla 45. Diseño controlador AccountController

Controlador	SlideController	
Atributos	\$scope.user	Usuario del sistema identificado mediante un identificador
Funciones	currentUser	Esta función obtiene la información del usuario
	subscribeResources	Esta función suscribe los recursos a los usuarios
	logout	Esta función cierra la sesión del usuario

Tabla 46. Diseño controlador SlideController

Interfaces

En este apartado se muestran los avances en el diseño a lo largo de las tres iteraciones realizadas. El diseño de la interfaz ha evolucionado a lo largo de la duración del proyecto debido al ciclo de vida incremental e iterativo elegido para la consecución del proyecto.

Iteración 1

A lo largo de esta primera iteración se presenta un *mockup* con las principales vistas del sistema para ser implementadas. Se busca un diseño minimalista, adaptado a la versión web del sistema y cuyo enfoque siga las líneas de diseño del ecosistema Android para móviles.

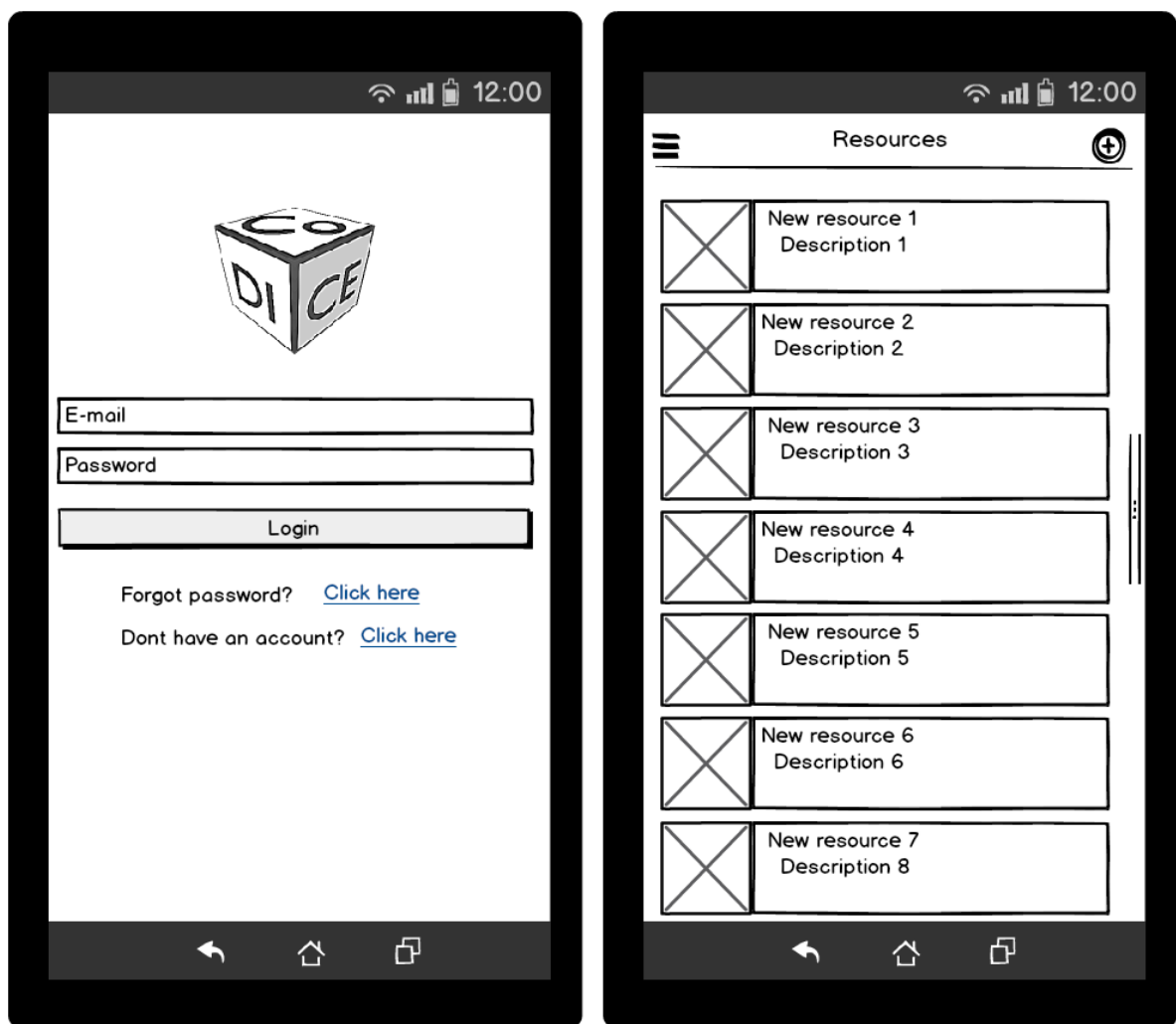


Ilustración 18. Mockups primera iteración – Login y Resources

En la [Ilustración 18](#) se muestra un *mockup* de las pantallas de login y de la lista de recursos del sistema. En ellas se puede apreciar la búsqueda de un diseño orientado a la funcionalidad evitando añadir elementos que puedan distraer al usuario. De este modo, en la pantalla de recursos se muestra la lista completa de recursos con la vista previa de la imagen, el nombre de cada recurso y la descripción de los mismos sin que exista ningún otro elemento que pueda destacar sobre los recursos.

A lo largo de la primera iteración también se presentan los *mockups* para el alta de usuario, recuperación de contraseña y menú de añadir recursos. Estas acciones se muestran en la [Ilustración 19](#).



Ilustración 19. Mockups primera iteración – Sign-Up, Pass Recovery y Add Resource

Todos estos diseños son implementados dentro de la primera iteración cumpliendo con el [Modelo del proceso](#) elegido para el proyecto.

Iteración 2

Durante la segunda iteración se realizan los diseños para la vista que muestra un único recurso, la interfaz de la cuenta y el panel del menú izquierdo de la aplicación. Estos diseños se pueden observar en la [Ilustración 20](#).

En la pantalla asociada al recurso se presenta el nombre de este (puede ser modificado por el usuario) y la imagen asociada al recurso o una vista previa si no se trata de una fotografía. En la parte inferior se encuentra la zona dedicada a las acciones de añadir información. Así, se puede observar un espacio para la descripción del recurso y un espacio para añadir comentarios.

La vista de la cuenta de usuario refleja los datos del usuario de la aplicación. Estos datos se pueden actualizar introduciendo la contraseña actual.

El último diseño corresponde con la barra lateral izquierda. Esta barra tiene por objeto mostrar un menú de accesos a distintas secciones de la aplicación. El diseño se ha realizado teniendo en cuenta las recomendaciones de diseño para dispositivos Android.

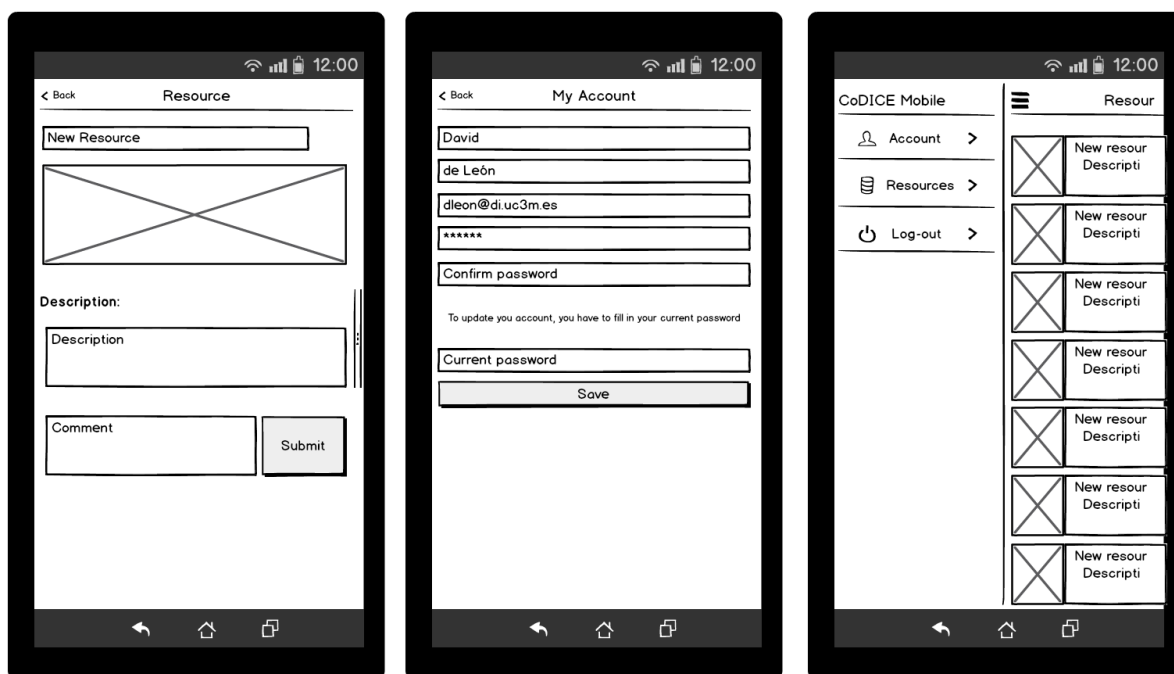


Ilustración 20. Mockups segunda iteración – Resource, My Account y left navigation

Del mismo modo que en la primera iteración, estos *mockups* son implementados y presentados al cliente en la reunión correspondiente a la finalización de la segunda iteración.

Iteración 3

La iteración 3 es la última iteración realizada en el desarrollo del proyecto. A nivel de diseño se trabaja sobre los prototipos funcionales ya implementados aunque de forma previa se continúa trabajando con *mockups* para obtener una visión previa a la implementación final de la solución. De este modo, se presenta un nuevo diseño de la vista encargada de mostrar los recursos de forma individual.

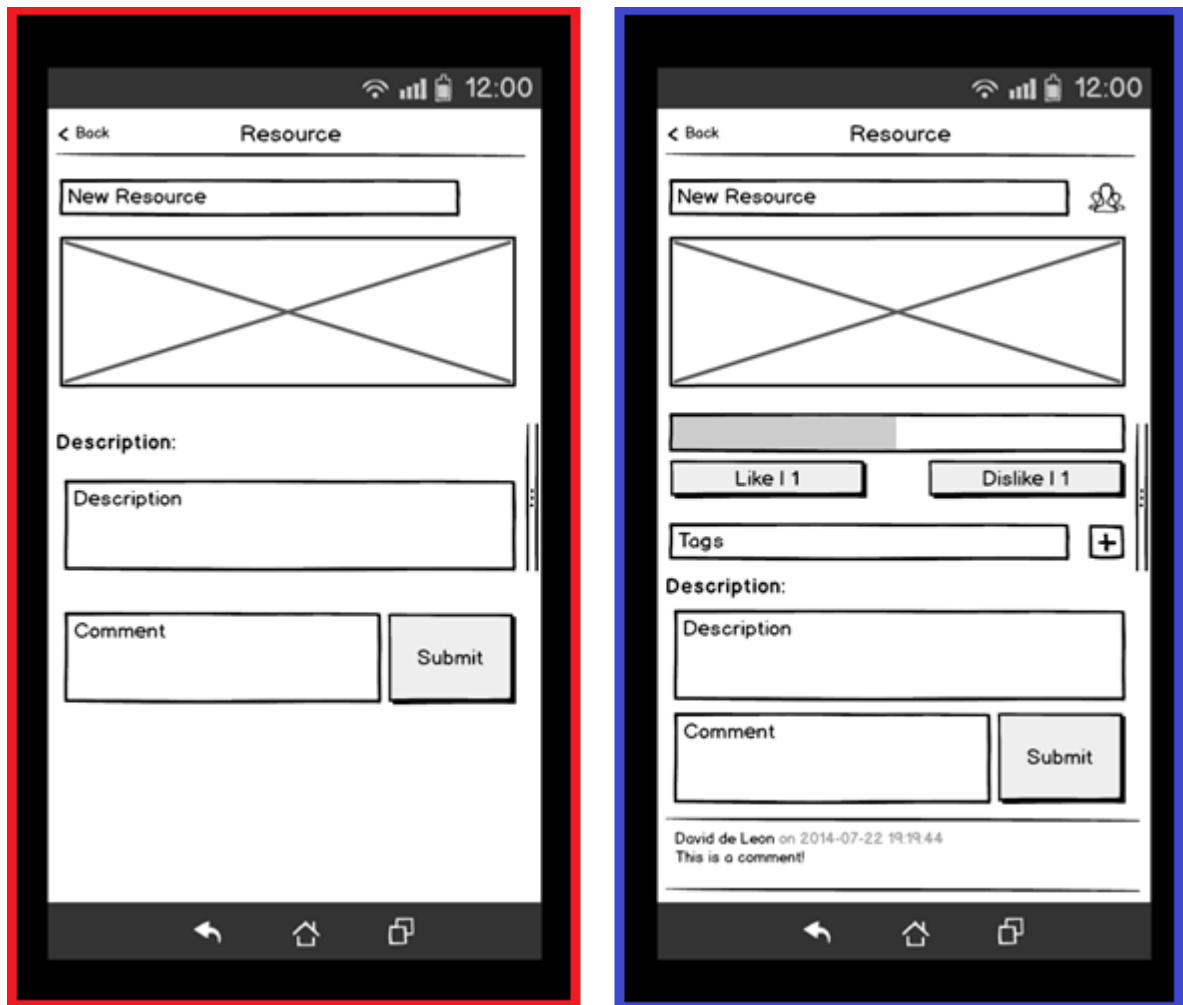


Ilustración 21. Mockups iteración – Comparativa de versiones para Resource

En la [Ilustración 21](#) se observa a la izquierda la versión implementada en la iteración anterior y a la derecha el nuevo diseño. Este nuevo diseño añade diferentes características:

- **Compartir recurso.** A la derecha del nombre del recurso se muestra un botón que permite compartir un recurso privado del usuario. Una vez comparte el recurso, ese botón desaparece de la vista del sistema.
- **Votación de recursos.** Debajo de la fotografía se añade una barra de progreso y dos botones que permiten votar a los usuarios si les gusta el recurso o no. La barra de progreso se moverá a la derecha (verde) o izquierda (rojo) según el voto.
- **Tags.** En este nuevo diseño se permite a los usuarios ver y añadir nuevas etiquetas a un recurso.

Además de este cambio, para la entrega final del proyecto se presenta una versión para dispositivos con diferentes dimensiones de pantalla, correspondiendo el diseño a la versión tablet del producto. No se han realizado mockups de esta versión tablet debido a que el diseño se ha basado el uso de tecnologías adaptables (*responsive design*) permitiendo que la vista del producto se ajuste al tamaño de pantalla.

Las figuras correspondientes con las vistas del producto final ya implementadas en su versión móvil y tablet se encuentran en el [Anexo VI. Prototipo final](#).

Implementación

En esta sección se realiza una presentación de la organización del proyecto. A lo largo del apartado se va a tratar el entorno de desarrollo, el sistema de archivos y el despliegue del proyecto.

Entorno de desarrollo

Intel® XDK es una plataforma de desarrollo creada por Intel® y basada en el editor web Brackets. Este entorno ofrece una herramienta centralizada que permite desarrollar, emular y depurar las aplicaciones de una forma rápida y eficaz.

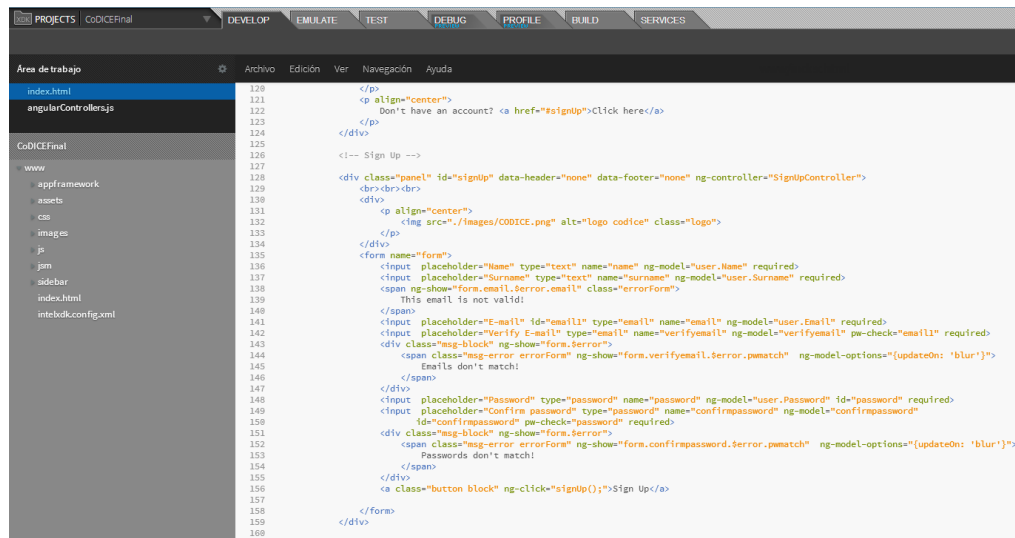


Ilustración 22. Pantalla de desarrollo de Intel® XDK

El IDE incluye por defecto la librería AppFramework utilizada en este proyecto para crear las interfaces del sistema. Asimismo, la plataforma permite integrar de un modo sencillo el resto de librerías necesarias para el proyecto.

A la hora de implementar la solución Intel® XDK ofrece la posibilidad de emular el comportamiento de la aplicación desde el propio entorno. Si se desea acceder a características del dispositivo como puede ser la cámara de fotos, la herramienta permite a su vez comprobar la funcionalidad desde un dispositivo móvil que esté conectado mediante puerto USB al ordenador de desarrollo. Todas estas características ofrecen grandes ventajas al desarrollador, el cual puede ir observando y probando los cambios sin necesidad de construir la aplicación.

Por último, es necesario destacar que Intel® XDK ofrece la posibilidad de generar la aplicación para distintas plataformas.

Estructura de archivos

El proyecto generado por Intel XDK ofrece una estructura de carpetas similar a la que se encuentra en un proyecto de desarrollo web con HTML5. En la [Ilustración 23](#) se muestran los archivos de configuración del proyecto generados de forma automática por el entorno de desarrollo y una carpeta donde se encuentran todos los archivos de la solución.

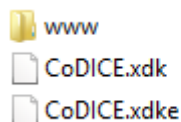


Ilustración 23. Estructura de archivos de AppFramework

En la [Ilustración 24](#) se presenta el sistema de archivos interno del proyecto.

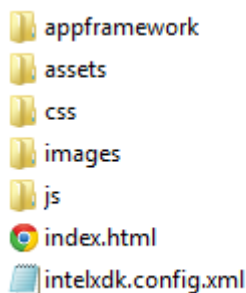


Ilustración 24. Estructura de archivos del proyecto CoDICE

La primera carpeta que se observa es la que incluye toda la librería AppFramework. Esta API viene cargada por defecto en el IDE para la creación de interfaces tal y como se ha explicado en el [Entorno de desarrollo](#).

En la carpeta *assets* se integran las imágenes que se muestran durante la primera carga del sistema. Esas imágenes son el logotipo de CoDICE en diversas dimensiones para adecuarse a los distintos tamaños de pantalla.

En la tercera carpeta se encuentran los CSS de la aplicación y las imágenes asociados a estas. Estos ficheros se presentan en la [Ilustración 25](#). La hoja de estilos *codice.css* cuenta con los estilos creados durante el desarrollo mientras que fichero *jquery.mobile-1.4.2.min.css* carga los estilos por defecto de JQuery Mobile.

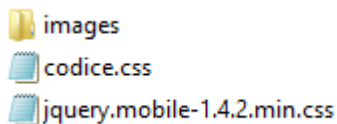


Ilustración 25. Archivos CSS

En la carpeta de imágenes se cargan fotografías del logotipo de CoDICE y otro tipo de ilustraciones que se han utilizado y no se encuentran presentes en la carpeta de estilos CSS.

La información que contiene la carpeta de JavaScript se presenta en la [Ilustración 26](#).

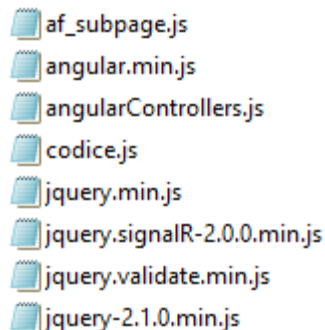


Ilustración 26. Archivos JavaScript

El fichero angularController.js implementa la capa controlador del sistema de la forma que ya se ha detallado en el [Diseño del Controlador](#). Por su parte, el fichero codice.js cuenta con el código que permite la conexión *Hub* entre el cliente y servidor. El resto de ficheros pertenecen a las diferentes librerías utilizadas en el proyecto y no han sido modificadas.

Despliegue de la aplicación

Para realizar el despliegue de la aplicación se ha utilizado la opción de construcción de la aplicación mediante el modo Cordova for Android que ofrece el entorno de desarrollo Intel® XDK. En la [Ilustración 27](#) se puede observar las diferentes opciones que proporciona la IDE para la creación del producto adaptado a distintos sistemas.

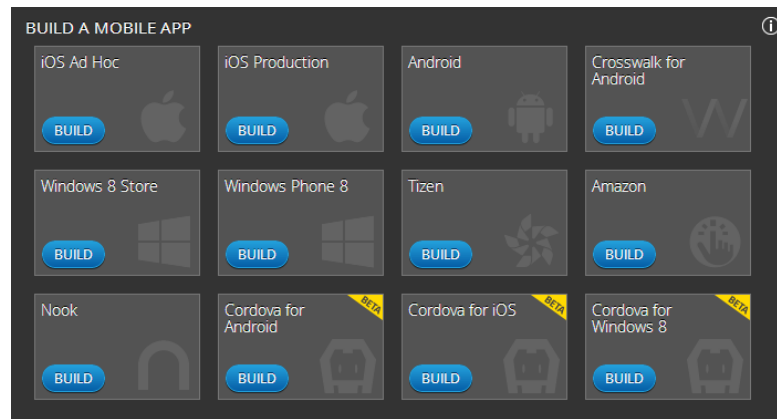


Ilustración 27. Pantalla de construcción de aplicación de Intel® XDK

Una vez elegida la plataforma, se sube el código de forma automática a los servidores de Intel® donde se realiza el ensamblado del producto. Una vez finalizado, el entorno permite la descarga del ejecutable de la aplicación. Si se desea, la herramienta ofrece la posibilidad de publicar la aplicación en las tiendas de cada una de las plataformas.

5 Evaluación

A lo largo de esta sección se realiza un estudio pormenorizado de la solución para verificar su validez frente a la [Definición del problema](#) y los [Objetivos](#) definidos. Para ello, se presenta un proceso de evaluación basado en la presentación de un plan de pruebas y el análisis de los resultados obtenidos.

5.1 Proceso de evaluación

Para el proceso de evaluación se trabaja en un plan de pruebas relacionado con los [Casos de uso](#) previamente especificados. La trazabilidad entre casos de prueba, casos de uso y requisitos del sistema permiten la validación del sistema.

Plan de pruebas

En esta sección se expone el plan de pruebas definido para la evaluación del producto. La [Tabla 47](#) presenta la plantilla a utilizar para dicho plan:

Caso de prueba	Identificador
Caso de uso asociado	
Precondiciones	
Pasos a seguir	
Verificación	

Tabla 47. Plantilla para casos de prueba

El significado de cada uno de los apartados de la plantilla se describe a continuación:

- **Caso de prueba.** Descripción breve del caso de prueba.
- **Identificador.** Muestra la identificación unívoca del caso de prueba. Este atributo está compuesto por tres letras seguidas por un guion y una serie de dos dígitos autoincrementales:
 - CPR-XX (Caso de Prueba)–(Digito1Digito2)
- **Caso de uso asociado.** Identificador del caso de uso al que se asocia el caso de prueba.
- **Precondiciones.** Condiciones previas necesarias a la ejecución del caso de prueba.

- **Pasos a seguir.** Describe de forma detallada los pasos a seguir para llevar a cabo el proceso asociado al caso de prueba.
- **Verificación.** Descripción del proceso que verifica la superación del caso de prueba.

Casos de pruebas

A continuación, se presenta la definición de los casos de prueba para la validación del proyecto. La especificación detallada se encuentra disponible en el [Anexo V. Especificación de casos de prueba](#).

- **CPR-01.** Alta de usuario en el sistema.
- **CPR-02.** Acceso de usuario a la aplicación.
- **CPR-03.** Recuperación de contraseña.
- **CPR-04.** Acceso a un recurso.
- **CPR-05.** Actualización del nombre del recurso.
- **CPR-06.** Actualización de la descripción del recurso.
- **CPR-07.** Nuevo comentario en un recurso.
- **CPR-08.** Evaluación positiva de un recurso.
- **CPR-09.** Evaluación negativa de un recurso.
- **CPR-10.** Asignación de etiquetas a un recurso.
- **CPR-11.** Creación de nueva etiquetas.
- **CPR-12.** Nuevo recurso de tipo fotografía.
- **CPR-13.** Nuevo recurso de tipo video.
- **CPR-14.** Nuevo recurso de tipo URI.
- **CPR-15.** Nuevo recurso tipo multimedia del dispositivo.
- **CPR-16.** Añadir información de nuevo recurso.
- **CPR-17.** Compartir recurso privado.
- **CPR-18.** Modificación de datos de acceso del usuario.

Matriz de trazabilidad

En la [Tabla 48](#) se muestra la matriz de trazabilidad entre los casos de uso y los casos de prueba.

	CU-01	CU-02	CU-03	CU-04	CU-05	CU-06	CU-07	CU-08	CU-09	CU-10	CU-11	CU-12	CU-13	CU-14	CU-15	CU-16	CU-17	CU-18
CPR-01	x																	
CPR-02		x																
CPR-03			x															
CPR-04				x														
CPR-05					x													
CPR-06						x												
CPR-07							x											
CPR-08								x										
CPR-09									x									
CPR-10										x								
CPR-11											x							
CPR-12												x						
CPR-13													x					
CPR-14														x				
CPR-15															x			
CPR-16																x		
CPR-17																	x	
CPR-18																		x

Tabla 48. Matriz de trazabilidad CU-CPR

5.2 Análisis de resultados

Una vez realizado el proceso de evaluación, se presentan los resultados de cada caso de prueba en la [Tabla 49](#).

Caso de prueba	Resultado
CPR-01	Satisfactorio
CPR-02	Satisfactorio
CPR-03	Satisfactorio
CPR-04	Satisfactorio
CPR-05	Satisfactorio
CPR-06	Satisfactorio
CPR-07	Satisfactorio
CPR-08	Satisfactorio
CPR-09	Satisfactorio
CPR-10	Satisfactorio
CPR-11	Satisfactorio
CPR-12	Satisfactorio
CPR-13	Satisfactorio
CPR-14	Satisfactorio
CPR-15	Satisfactorio
CPR-16	Satisfactorio
CPR-17	Satisfactorio
CPR-18	Satisfactorio

Tabla 49. Análisis de resultados de los casos de prueba

El análisis de resultados de los casos de prueba presenta un resultado satisfactorio para todos ellos. Este hecho, unido a la trazabilidad de las pruebas con respecto a los [Casos de uso](#) permite afirmar que el sistema cumple con los objetivos especificados del proyecto y confirma la validez de la solución presentada.

6 Conclusiones

En esta sección del documento se exponen las conclusiones obtenidas tras la realización del proyecto. Se recogen las aportaciones realizadas, los trabajos futuros y los problemas encontrados durante la realización del mismo. Para finalizar, se incluyen las opiniones personales del autor de este proyecto.

6.1 Aportaciones realizadas

La principal aportación a este proyecto ha sido el diseño, desarrollo y puesta en funcionamiento de una aplicación móvil multiplataforma para la obtención y gestión de recursos dentro del contexto cultural al que pertenece CoDICE. Con la creación de la aplicación móvil se ha conseguido ofrecer al usuario un método rápido y sencillo para obtener y compartir recursos sobre los que trabajar en procesos de diseño colaborativo.

A su vez, se ha conseguido integrar el sistema móvil con la aplicación web ya desarrollada logrando extender la funcionalidad del sistema inicial mediante la creación de un sistema distribuido en tiempo real.

Por último, se han creado las bases para la obtención de nuevas versiones del producto adecuadas a las distintas plataformas móviles, gracias a la labor de investigación y elección de tecnologías para el desarrollo del producto.

6.2 Trabajos futuros

Las líneas de trabajo futuro para este proyecto deben estar orientadas a cuestiones de usabilidad y diseño.

En primer lugar, es recomendable realizar una evaluación con expertos en diseño y usabilidad de sistemas móviles para determinar las posibles deficiencias con las que cuente el sistema en estas dos facetas. Tras esta evaluación, se debe presentar la aplicación y realizar una nueva evaluación con los usuarios finales del sistema, de modo que se pueda obtener un *feedback* directo de las personas que van a utilizar la aplicación a diario.

A continuación, se debería plantear como trabajo futuro la adaptación de las interfaces a los sistemas operativos móviles iOS y Windows Phone. Las decisiones tecnológicas y el patrón de

desarrollo utilizado ofrecen la posibilidad de realizar estas versiones en un periodo muy corto de tiempo.

Por otra parte, resulta interesante valorar la evolución de la base de usuarios del sistema y estudiar si sería recomendable la creación de una aplicación nativa para algunas de las principales plataformas móviles. A la hora de realizar este proyecto, se detectó que los usuarios contaban con dispositivos de todas las plataformas del mercado por lo que fue un requisito indispensable poder llegar a todos ellos de una forma rápida y eficaz. Aun así, es necesario tener presente la rápida evolución de la tecnología y del mercado que puede provocar un cambio de tendencias en un periodo muy corto de tiempo. Por ello, ante futuras versiones del sistema se recomienda realizar un estudio pormenorizado de la situación del mercado y del público objetivo al que va dirigida la aplicación.

6.3 Problemas encontrados

Es necesario destacar que a pesar de que el desarrollo del proyecto se ha llevado a cabo con éxito y se han cumplido los objetivos, el camino para conseguirlo no ha estado exento de dificultades:

- **Desconocimiento del codiseño.** A la hora de afrontar el proyecto la mayor problemática fue el no entender las bases del funcionamiento del proyecto CoDICE debido a la falta de referencias sobre las técnicas de codiseño. Para lograr superarlo, fue necesario realizar un estudio pormenorizado sobre codiseño y las técnicas que se aplican. De forma paralela al estudio teórico, estuve trabajando a diario con las opciones de la versión web de CoDICE para familiarizarme con su uso y poder extraer los objetivos y necesidades requeridas para la aplicación móvil.
- **Falta de experiencia en el desarrollo móvil.** El conocimiento previo de las tecnologías móviles con las que se desarrolló el producto era muy limitado por mi parte para afrontar un proyecto con unas necesidades tan concretas. A lo largo de la implementación del producto fue necesario realizar gran cantidad de pruebas de concepto para distintas fases, sobre todo a nivel de desarrollo con lenguaje JavaScript.

6.4 Opiniones personales

La realización de este proyecto ha supuesto diversos retos a nivel personal y académico. El principal de los retos ha sido enfrentarme a la creación de un producto que va a formar parte de un sistema en el que su versión web ya se encuentra accesible a usuarios. Hasta ahora, todos los proyectos en los que he trabajado han sido puramente académicos, y aunque éste forma parte de una línea de investigación, es el primero que llegará al usuario final.

Otra de las cuestiones a destacar fue el estudio de alternativas tecnológicas. Una elección de tecnologías errónea no hubiese permitido la consecución de los objetivos marcados. Para evitarlo, el proceso de estudio se alargó en el tiempo más de lo que estaba planificado. A posteriori, dedicar más tiempo a analizar las opciones se puede considerar como una decisión acertada, ya que es uno de los motivos principales por los que el producto finalizó cumpliendo los objetivos en la fecha planeada.

Durante este periodo, he adquirido grandes conocimientos sobre organización entre distintos grupos de trabajo involucrados en un mismo proyecto. De este modo, ha sido muy enriquecedor mantener reuniones diarias y discusiones tecnológicas con el fin de poder conseguir la comunicación entre el cliente móvil, el cliente web y el servidor de aplicaciones.

Por último, considero necesario destacar los numerosos conocimientos técnicos adquiridos con la diversidad de tecnologías utilizadas y la aplicación de los conocimientos sobre ingeniería del software en el desarrollo de un proyecto real.

7 Bibliografía

- [1] B. -N. Sanders y P. Stappers, «Cocreation and the new landscapes of design,» *CoDesign: International Journal of CoCreation in Design and the Arts*, pp. 5-18, 2008.
- [2] «<http://mesch-project.eu/>,» [En línea]. Available: <http://mesch-project.eu/>. [Último acceso: 25 08 2014].
- [3] P. D. Pérez, «<http://mesch-project.eu/>,» 26 07 2014. [En línea]. Available: <http://mesch-project.eu/codice-using-software-tools-to-support-co-design-activities/>. [Último acceso: 26 08 2014].
- [4] «servicedesigntools.org,» [En línea]. Available: <http://servicedesigntools.org/taxonomy/term/2>. [Último acceso: 10 Mayo 2014].
- [5] «www.mindtools.com,» [En línea]. Available: http://www.mindtools.com/pages/article/newTMC_86.htm. [Último acceso: 23 Junio 2014].
- [6] «http://www.instructionaldesign.org,» [En línea]. Available: <http://www.instructionaldesign.org/storyboarding.html>. [Último acceso: 8 Agosto 2014].
- [7] «<http://think.withgoogle.com/mobileplanet/es/>,» 2013. [En línea]. Available: <http://services.google.com/fh/files/misc/omp-2013-es-local.pdf>. [Último acceso: 9 agosto 2014].
- [8] D. Matthew, HTML5: designing rich Internet applications, Focal Press, 2013.
- [9] «http://www.volere.co.uk,» [En línea]. Available: http://www.volere.co.uk/pdf%20files/template_es.pdf. [Último acceso: 22 agosto 2014].
- [10] «<https://angularjs.org/>,» [En línea]. Available: <https://angularjs.org/>. [Último acceso: 3 Agosto 2014].

- [11] «<http://www.lab.inf.uc3m.es>,» [En línea]. Available:
<http://www.lab.inf.uc3m.es/~a0080802/RAI/mvc.html>. [Último acceso: 23 agosto 2014].
- [12] «<http://cordova.apache.org/>,» [En línea]. Available:
<http://cordova.apache.org/docs/en/3.5.0/>. [Último acceso: 3 agosto 2014].
- [13] «<http://www.asp.net>,» [En línea]. Available:
<http://www.asp.net/signalr/overview/signalr-20/hubs-api/hubs-api-guide-server>.
[Último acceso: 20 agosto 2014].
- [14] «<http://cordova.apache.org>,» [En línea]. Available:
http://cordova.apache.org/docs/en/2.5.0/cordova_file_file.md.html. [Último acceso: 22 julio 2014].
- [15] «<http://www.antoniorigo.com>,» [En línea]. Available:
<http://www.antoniorigo.com/disenio-web-adaptable/>. [Último acceso: 20 julio 2014].

Anexo I. Control de versiones

Versión	Fecha	Descripción
0.1	13/06/14	Redacción del capítulo 1: “Introducción”
0.2	02/08/14	Redacción del capítulo 2: “El estado de la cuestión” sección “Revisión de procesos de codiseño”
0.2.1	06/08/14	Redacción del capítulo 2: “El estado de la cuestión” sección “Revisión de procesos de codiseño”
0.2.2	07/08/14	Redacción del capítulo 2: “El estado de la cuestión” secciones “Revisión de arquitecturas móviles”, “Revisión de tecnologías móviles”
0.2.3	08/08/14	Redacción del capítulo 2: “El estado de la cuestión” sección “Resumen del estado de la cuestión”
0.3	15/08/14	Redacción del capítulo 3: “Gestión del proyecto software” secciones “Alcance del Proyecto” y “Estimación de recursos”
0.3.1	17/08/14	Redacción del capítulo 3: “Gestión del proyecto software” sección “Presupuesto”
0.4	23/08/14	Redacción del capítulo 4: “Solución” sección “Descripción de la solución”
0.4.1	25/08/14	Redacción del capítulo 4: “Solución” sección “El proceso de desarrollo” apartados “Modelo de proceso” y “Análisis”
0.4.2	28/08/14	Redacción del capítulo 4: “Solución” sección “El proceso de desarrollo” apartados “Diseño” e “Implementación”
0.5	29/08/14	Redacción del capítulo 5: “Evaluación”
1.0	01/09/14	Revisión parcial del documento
1.1	04/09/14	Redacción del capítulo 6: “Conclusión”

2.0	06/09/14	Revisión parcial del documento
3.0	14/09/14	Revisión final del documento

Tabla 50. Control de versiones

Anexo II. Seguimiento de Trabajo Fin de Grado

A lo largo de este anexo se presenta el seguimiento realizado durante la duración de este proyecto.

Forma de seguimiento

Para la realización del seguimiento se acordó con el tutor la realización de varias reuniones en las que presentó la evolución de la implementación y la documentación asociada al proyecto. Para ello, se ha utilizado la herramienta Microsoft Project 2013.

Planificación inicial

La planificación inicial recoge una duración del proyecto de 4 meses en la que se estima la duración de cada una de las fases del proyecto. Asimismo, se han diferenciado las tareas de creación del producto con las de la elaboración del documento asociado a éste. En la [Ilustración 28](#) se muestra la planificación inicial con las tareas presentadas en el apartado [Plan de trabajo](#).

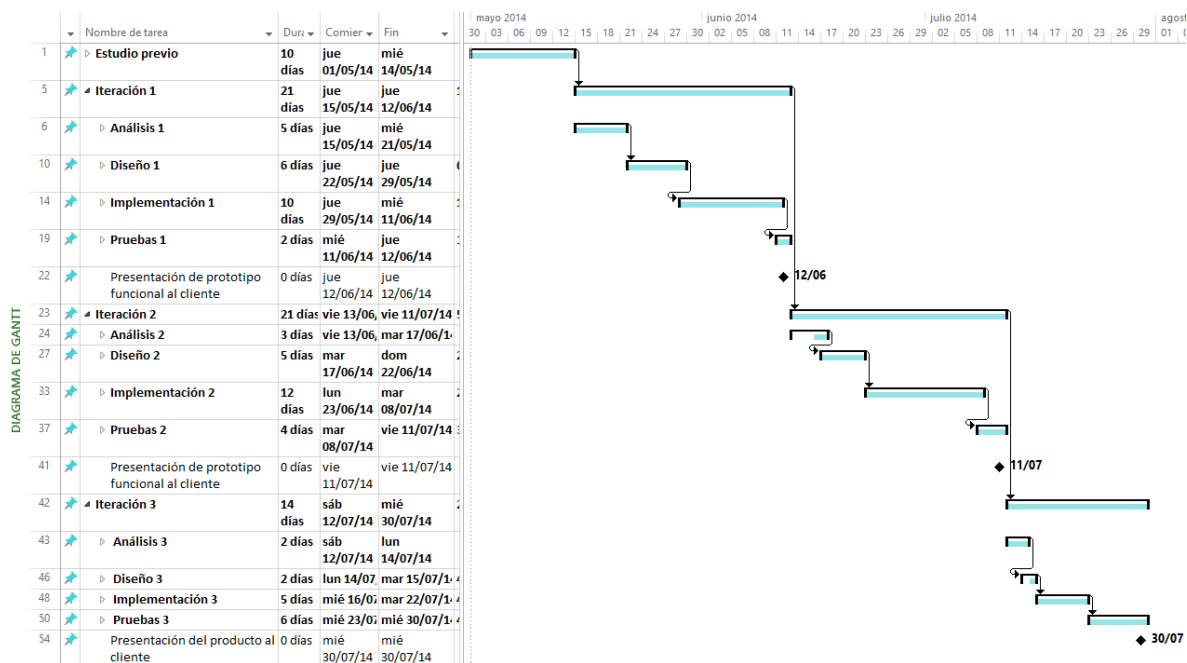


Ilustración 28. Gantt - Planificación inicial del desarrollo

La planificación inicial para asociada a la documentación del documento se muestra en la **Ilustración 29**:

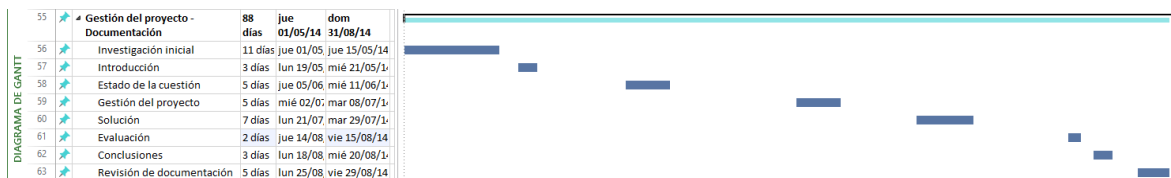


Ilustración 29. Gantt - Planificación inicial de la documentación

Planificación final

La planificación final recoge la duración real de cada una de las tareas asociadas al proyecto. Estos datos se muestran en la **Ilustración 30**:

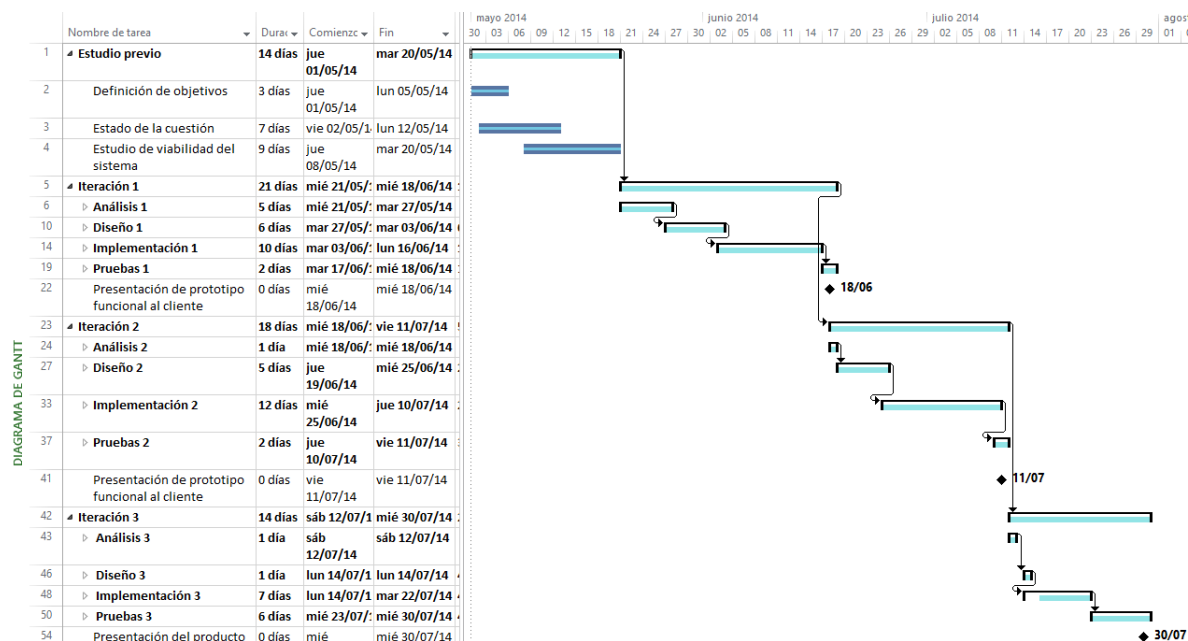


Ilustración 30. Gantt - Planificación final del desarrollo

Por su parte, la planificación final asociada a la documentación del proyecto se presenta en la **Ilustración 31**:

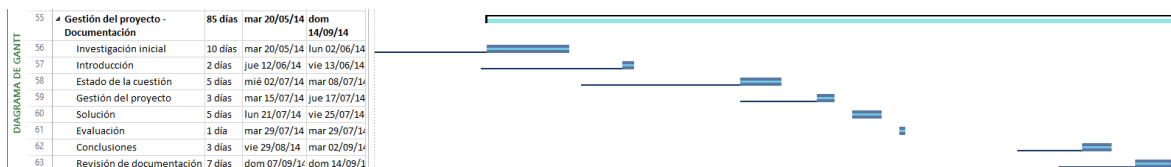


Ilustración 31. Gantt - Planificación final de la documentación

Estudio del seguimiento

La presentación del sistema final se realizó en la fecha acordada, el 30 de julio de 2014. A pesar de ello, se presentaron diversas modificaciones en la planificación para algunas de las tareas. En concreto, la duración del estudio previo aumento en cuatro días. Este tiempo se recuperó a lo largo de las tareas de análisis y diseño correspondientes a las iteraciones 2 y 3. En estas tareas se produjo la mayor variación entre planificación inicial y final debido a una sobreestimación del tiempo inicial.

Por su parte, la documentación sufrió un retraso de catorce días con respecto a la planificación. La elaboración del documento comenzó con un retraso de dos semanas y volvió a ser pospuesto debido a la carga de trabajo requerida por la elaboración del proyecto y la cercanía de los exámenes finales. Tras retomar el trabajo de documentación, se comprobó que la planificación inicial era muy conservadora y aunque no se consiguió cumplir con la fecha final establecida, la duración total de la elaboración del documento se realizó en 27 días en vez de en los 35 inicialmente planificados.

Anexo III. Especificación de requisitos

Requisitos funcionales

Identificador	RF-01		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario darse de alta mediante la introducción de nombre, apellido, correo electrónico y contraseña		
Prueba de verificación	El usuario será capaz de darse de alta en la aplicación rellenando todos los campos solicitados		

Identificador	RF-02		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario acceder a la aplicación con su dirección de correo electrónico y contraseña		
Prueba de verificación	El usuario será capaz de acceder a la aplicación rellenando los campos solicitados		

Identificador	RF-03		
Prioridad	Baja	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario recuperar su contraseña mediante su dirección de correo electrónico		
Prueba de verificación	El usuario será capaz de recuperar su contraseña rellenando el campo solicitado		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RF-04		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema mostrará al usuario sus recursos privados en modo lista		
Prueba de verificación	El usuario será capaz de visualizar los recursos que no ha compartido con el resto de usuarios		

Identificador	RF-05		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema mostrará al usuario los recursos compartidos por el resto de usuarios en modo lista		
Prueba de verificación	El usuario será capaz de visualizar los recursos compartidos de la aplicación		

Identificador	RF-06		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema permitirá al usuario acceder a un recurso de la lista visualizando todos los datos del mismo		
Prueba de verificación	El usuario será capaz de visualizar un recurso tras realizar la acción de acceder al mismo		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RF-07		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario modificar el nombre del recurso de manera instantánea		
Prueba de verificación	El usuario será capaz de visualizar el cambio realizado por él mismo en el nombre del recurso		

Identificador	RF-08		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario modificar la descripción del recurso de manera instantánea		
Prueba de verificación	El usuario será capaz de visualizar el cambio realizado en la descripción por él mismo en la descripción del recurso		

Identificador	RF-09		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario comentar recursos		
Prueba de verificación	El usuario será capaz de visualizar su comentario una vez lo haya añadido con el botón correspondiente		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RF-10		
Prioridad	Media	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario evaluar recursos mediante las opciones me gusta y no me gusta		
Prueba de verificación	El usuario será capaz de visualizar su voto tras haberlo realizado		

Identificador	RF-11		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario asignar etiquetas (tags) por defecto a los recursos		
Prueba de verificación	El usuario será capaz de elegir entre las etiquetas ya creadas por defecto y que ésta se asigne al recurso		

Identificador	RF-12		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario crear nuevas etiquetas (tags) y asignarlas a los recursos		
Prueba de verificación	El usuario será capaz de añadir una nueva etiqueta y visualizarla en la lista con el resto		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RF-13		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema permitirá al usuario capturar y subir fotografías como nuevos recursos		
Prueba de verificación	El usuario será capaz de acceder a la cámara de su dispositivo y realizar una fotografía. Esta fotografía será mostrada en la pantalla de recurso para que pueda añadir los campos de nombre y descripción		

Identificador	RF-14		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema permitirá al usuario capturar y subir videos como nuevos recursos		
Prueba de verificación	El usuario será capaz de acceder a la cámara de su dispositivo y realizar un video. La aplicación redirigirá al usuario al menú de recurso para que pueda añadir los campos de nombre y descripción		

Identificador	RF-15		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema permitirá al usuario añadir y subir URIs como nuevos recursos		
Prueba de verificación	El usuario será capaz de añadir una URI. La aplicación redirigirá al usuario al menú de recurso para que pueda añadir los campos de nombre y descripción		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RF-16		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema permitirá al usuario añadir contenido multimedia alojado en el móvil como nuevos recursos		
Prueba de verificación	El usuario será capaz de acceder a las carpetas de su dispositivo y elegir un contenido multimedia. La aplicación redirigirá al usuario al menú de recurso para que pueda añadir los campos de nombre y descripción		

Identificador	RF-17		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario modificar sus datos de acceso		
Prueba de verificación	El usuario será capaz de visualizar sus datos de acceso modificados		

Identificador	RF-18		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá al usuario compartir los recursos privados mediante el botón compartir		
Prueba de verificación	Los usuarios serán capaces de visualizar el recurso compartido		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RF-19		
Prioridad	Alta	Tipo de Requisito	Usuario
Descripción	El sistema permitirá que las actualizaciones de los recursos se vean de forma inmediata por el resto de usuarios		
Prueba de verificación	El usuario será capaz de visualizar los cambios realizados por el resto de usuario a la vez que estos se producen		

Requisitos no funcionales

Identificador	RNF-01		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema será desarrollado para un tamaño de pantalla mínimo de 4 pulgadas		
Prueba de verificación	El sistema se verá de forma correcta en dispositivos con una pantalla de 4 o más pulgadas		

Identificador	RNF-02		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	El sistema será creado con tecnologías web		
Prueba de verificación	-		

UNIVERSIDAD CARLOS III DE MADRID

Identificador	RNF-03		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	Las actualizaciones de recursos deben ser visibles por los usuarios en tiempo real		
Prueba de verificación	El usuario deberá recibir la actualización de un recurso por parte de otro usuario en tiempo real		

Identificador	RNF-04		
Prioridad	Media	Tipo de Requisito	Sistema
Descripción	La interfaz será adaptada a los patrones de diseño de cada plataforma móvil		
Prueba de verificación	-		

Identificador	RNF-05		
Prioridad	Alta	Tipo de Requisito	Sistema
Descripción	Las tecnologías del sistema deberán ser compatibles con ASP.NET y Microsoft Silverlight		
Prueba de verificación	-		

Anexo IV. Especificación de casos de uso

Caso de uso	Alta de usuario	Identificador	CU-01
Actores	Usuario no registrado en la aplicación		
Escenario	El usuario desea darse de alta en la aplicación móvil		
Descripción	El usuario accede a la aplicación desde su dispositivo móvil y se da de alta en el sistema		
Precondiciones	El usuario debe tener un dispositivo móvil La aplicación debe estar instalada en el dispositivo El usuario no debe estar dado de alta		
Pasos a seguir	1. El usuario entra en la aplicación 2. El usuario accede al menú que le permite darse de alta 3. El usuario rellena los campos obligatorios 4. El usuario pulsa el botón para darse de alta en la aplicación		

Caso de uso	Acceso a la aplicación	Identificador	CU-02
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea acceder a la aplicación móvil		
Descripción	El usuario accede al sistema		
Precondiciones	El usuario debe estar dado de alta en el sistema El usuario no debe estar logueado en el sistema		
Pasos a seguir	1. El usuario entra en la aplicación desde su dispositivo móvil 2. El usuario introduce su correo electrónico y contraseña 3. El usuario pulsa el botón para acceder al sistema (login)		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Recuperación de contraseña	Identificador	CU-03
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea recuperar su contraseña		
Descripción	El usuario introduce los datos que le permiten recuperar su contraseña		
Precondiciones	El usuario debe estar dado de alta en el sistema		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación desde su dispositivo móvil 2. El usuario accede al menú que le permite recuperar la contraseña 3. El usuario introduce su dirección de correo electrónico en el campo habilitado 4. El usuario pulsa el botón para habilitado para recuperar la contraseña 		

Caso de uso	Acceso a recurso	Identificador	CU-04
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea acceder a un recurso		
Descripción	El usuario accede a un recurso desde la lista de recursos		
Precondiciones	El usuario debe estar logueado en el sistema		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de la lista de recursos 2. El usuario elige un recurso de la lista 3. El usuario pulsa sobre el recurso elegido 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Actualización nombre del recurso	Identificador	CU-05
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea acceder actualizar el nombre del recurso		
Descripción	El usuario modifica el nombre de un recurso ya creado		
Precondiciones	El usuario debe estar logueado en el sistema		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario se encuentra dentro del recurso 2. El usuario modifica el nombre del recurso con el teclado del dispositivo 		

Caso de uso	Actualización de la descripción del recurso	Identificador	CU-06
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea acceder actualizar la descripción del recurso		
Descripción	El usuario modifica la descripción de un recurso ya creado		
Precondiciones	<p>El usuario debe estar logueado en el sistema</p> <p>El usuario se encuentra dentro del recurso</p>		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario modifica la descripción del recurso con el teclado del dispositivo 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Nuevo comentario en recurso	Identificador	CU-07
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea comentar un recurso		
Descripción	El usuario realiza un comentario de un recurso ya creado		
Precondiciones	El usuario debe estar logueado en el sistema		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario se encuentra dentro del recurso 2. El usuario introduce un comentario en el campo dedicado a ello con el teclado del dispositivo 3. El usuario pulsa el botón que permite añadir el comentario 		

Caso de uso	Evaluación positiva de un recurso	Identificador	CU-08
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea valorar de forma positiva un recurso		
Descripción	El usuario valora de forma positiva un recurso		
Precondiciones	<p>El usuario debe estar logueado en el sistema</p> <p>El usuario se encuentra dentro del recurso</p>		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa el botón <i>like</i> 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Evaluación negativa de un recurso	Identificador	CU-09
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea valorar de forma negativa un recurso		
Descripción	El usuario valora de forma negativa un recurso		
Precondiciones	El usuario debe estar logueado en el sistema El usuario se encuentra dentro del recurso		
Pasos a seguir	1. El usuario pulsa el botón <i>dislike</i>		

Caso de uso	Asignación de etiquetas a un recurso	Identificador	CU-10
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea asignar una etiqueta a un recurso		
Descripción	El usuario asigna una etiqueta ya creada a un recurso		
Precondiciones	El usuario debe estar logueado en el sistema El usuario se encuentra dentro del recurso La etiqueta ya existe en el sistema		
Pasos a seguir	1. El usuario pulsa sobre el campo <i>tags</i> 2. El usuario comienza a escribir la etiqueta que desea 3. El usuario elige del menú desplegable la etiqueta		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Creación de nuevas etiquetas	Identificador	CU-11
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea crear una nueva etiqueta y asignarla a un recurso		
Descripción	El usuario crea una etiqueta ya la asigna a un recurso		
Precondiciones	<p>El usuario debe estar logueado en el sistema</p> <p>El usuario se encuentra dentro del recurso</p> <p>La etiqueta no ha sido creada</p>		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el campo <i>tags</i> 2. El usuario escribe la nueva etiqueta 3. El usuario añade la nueva etiqueta mediante el botón asignado 		

Caso de uso	Nuevo recurso de tipo fotografía	Identificador	CU-12
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea añadir un nuevo recurso de tipo fotografía		
Descripción	El usuario realiza una foto con el dispositivo y la añade a recursos		
Precondiciones	<p>El usuario debe estar logueado en el sistema</p> <p>El usuario se encuentra dentro de la lista de recursos</p>		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de añadir recursos 2. El usuario selecciona la fotografía como nuevo recurso 3. El usuario realiza una foto con la cámara del dispositivo 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Nuevo recurso de tipo video	Identificador	CU-13
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea añadir un nuevo recurso de tipo video		
Descripción	El usuario realiza un video con el dispositivo y la añade a recursos		
Precondiciones	El usuario debe estar logueado en el sistema El usuario se encuentra dentro de la lista de recursos		
Pasos a seguir	1. El usuario pulsa sobre el botón de añadir recursos 2. El usuario selecciona video como nuevo recurso 3. El usuario realiza un video con la cámara del dispositivo		

Caso de uso	Nuevo recurso de tipo URI	Identificador	CU-14
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea añadir un nuevo recurso de tipo URI		
Descripción	El usuario añade una nueva URI a recursos		
Precondiciones	El usuario debe estar logueado en el sistema El usuario se encuentra dentro de la lista de recursos		
Pasos a seguir	1. El usuario pulsa sobre el botón de añadir recursos 2. El usuario selecciona añadir URI como nuevo recurso 3. El usuario escribe una nueva dirección URI con el teclado del dispositivo 4. El usuario pulsa el botón para añadir la URI como un nuevo recurso		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Nuevo recurso de tipo multimedia del dispositivo	Identificador	CU-15
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea añadir un nuevo recurso de tipo multimedia que ya se encuentra alojado en el dispositivo móvil		
Descripción	El usuario añade un nuevo recurso ya almacenado en su dispositivo móvil		
Precondiciones	El usuario debe estar logueado en el sistema El usuario se encuentra dentro de la lista de recursos		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón de añadir recursos 2. El usuario selecciona añadir nuevo archivo como recurso 3. El usuario selecciona el archivo a añadir desde la galería del dispositivo móvil 		

Caso de uso	Añadir información de nuevo recurso	Identificador	CU-16
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea añadir la información asociada al nuevo recurso añadido		
Descripción	El usuario añade información a un nuevo recurso subido al sistema		
Precondiciones	El usuario ha añadido un nuevo recurso El sistema ha navegado hasta la pantalla de nuevo recurso		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario introduce nombre al nuevo recurso 2. El usuario introduce descripción al nuevo recurso 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de uso	Compartir recurso privado	Identificador	CU-17
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea compartir un recurso marcado como privado		
Descripción	El usuario comparte un recurso privado		
Precondiciones	El usuario debe estar logueado en el sistema El usuario debe estar dentro del recurso El recurso es privado		
Pasos a seguir	1. El usuario pulsa el botón para compartir recursos		

Caso de uso	Modificación de datos de acceso del usuario	Identificador	CU-18
Actores	Usuario registrado en la aplicación		
Escenario	El usuario desea cambiar alguno de sus datos de acceso		
Descripción	El usuario cambia alguno de sus datos de acceso		
Precondiciones	El usuario debe estar logueado en el sistema El usuario debe estar en el menú de recursos		
Pasos a seguir	1. El usuario accede al menú lateral 2. El usuario accede a su cuenta 3. El usuario realiza cambios en la información que desee 4. El usuario introduce su contraseña 5. El usuario confirma su contraseña 6. El usuario pulsa el botón para realizar el cambio en sus datos de acceso		

Anexo V. Especificación de casos de prueba

Caso de prueba	Alta de usuario	Identificador	CPR-01
Caso de uso asociado	CU-01		
Precondiciones	<p>El usuario debe tener un dispositivo móvil</p> <p>La aplicación debe estar instalada en el dispositivo</p> <p>El usuario no debe estar dado de alta</p>		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario pulsa "Click here" en la opción "Don't have an account?" 3. El usuario introduce su nombre 4. El usuario introduce su apellido 5. El usuario introduce su correo electrónico 6. El usuario introduce de nuevo su correo electrónico 7. El usuario introduce la contraseña 8. El usuario introduce de nuevo la contraseña 9. El usuario pulsa el botón "Sign-up" 		
Verificación	<ul style="list-style-type: none"> - El usuario recibe un correo electrónico con los datos de acceso - El sistema muestra al usuario la pantalla de login 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Acceso a la aplicación	Identificador	CPR-02
Caso de uso asociado	CU-02		
Precondiciones	Se debe haber superado la prueba CPR-01		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario introduce su correo electrónico 3. El usuario introduce su contraseña 4. El usuario pulsa el botón “Login” 		
Verificación	<ul style="list-style-type: none"> - El usuario accede al sistema y la aplicación muestra la lista de recursos 		

Caso de prueba	Recuperación de contraseña	Identificador	CPR-03
Caso de uso asociado	CU-03		
Precondiciones	El usuario debe estar dado de alta en el sistema		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario pulsa “Click here” en la opción “Forgot password?” 3. El usuario introduce su correo electrónico 4. El usuario pulsa el botón “Password recovery” 		
Verificación	<ul style="list-style-type: none"> - El usuario recibe un correo electrónico con los datos de acceso - El sistema muestra al usuario la pantalla de login 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Acceso a un recurso	Identificador	CPR-04
Caso de uso asociado	CU-04		
Precondiciones	Se debe haber superado la prueba CPR-02		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario se encuentra en el menú de la lista de recursos 2. El usuario pulsa sobre un recurso de la lista 		
Verificación	<ul style="list-style-type: none"> - El sistema muestra al usuario el recurso 		

Caso de prueba	Actualización nombre del recurso	Identificador	CPR-05
Caso de uso asociado	CU-05		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario escribe con el teclado del dispositivo un nuevo nombre al recurso 		
Verificación	<ul style="list-style-type: none"> - El sistema actualiza el nombre del recurso de forma automática 		

Caso de prueba	Actualización de la descripción del recurso	Identificador	CPR-06
Caso de uso asociado	CU-06		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario escribe con el teclado del dispositivo una nueva descripción al recurso 		
Verificación	<ul style="list-style-type: none"> - El sistema actualiza la descripción del recurso de forma automática 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Nuevo comentario en recurso	Identificador	CPR-07
Caso de uso asociado	CU-07		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario escribe con el teclado del dispositivo un nuevo comentario al recurso 2. El usuario pulsa el botón “Submit” 		
Verificación	<ul style="list-style-type: none"> - El sistema añade y muestra el comentario introducido por el usuario en la lista de comentarios 		

Caso de prueba	Evaluación positiva de un recurso	Identificador	CPR-08
Caso de uso asociado	CU-08		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa el botón “Like” 		
Verificación	<ul style="list-style-type: none"> - El sistema actualiza el contador de votos añadiendo un voto <i>like</i> - El sistema actualiza la barra de evaluación de recursos 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Evaluación negativa de un recurso	Identificador	CPR-09
Caso de uso asociado	CU-09		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	1. El usuario pulsa el botón “Dislike”		
Verificación	<ul style="list-style-type: none"> - El sistema actualiza el contador de votos añadiendo un voto <i>dislike</i> - El sistema actualiza la barra de evaluación de recursos 		

Caso de prueba	Asignación de etiquetas a un recurso	Identificador	CPR-10
Caso de uso asociado	CU-10		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el campo de texto “Tags” 2. El usuario escribe el nombre de la etiqueta que desea en el campo de texto “Tags” 3. El usuario elige del menú que se despliega la etiqueta que desea 4. El usuario pulsa el botón “+” 		
Verificación	<ul style="list-style-type: none"> - El sistema añade la etiqueta al recurso 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Creación de nueva etiquetas	Identificador	CPR-11
Caso de uso asociado	CU-11		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el campo de texto “Tags” 2. El usuario escribe el nombre de la etiqueta que desea en el campo de texto “Tags” 3. El usuario pulsa el botón “+” 		
Verificación	<ul style="list-style-type: none"> - El sistema añade la etiqueta a la lista de etiquetas - El sistema añade la etiqueta al recurso 		

Caso de prueba	Nuevo recurso de tipo fotografía	Identificador	CPR-12
Caso de uso asociado	CU-12		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “+” situado en la parte superior derecha 2. El usuario pulsa “Take a picture” 3. El usuario realiza una foto con la cámara del dispositivo 		
Verificación	<ul style="list-style-type: none"> - El sistema carga la nueva fotografía - El sistema muestra la vista de recurso para que el usuario añada la información del nuevo recurso 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Nuevo recurso de tipo video	Identificador	CPR-13
Caso de uso asociado	CU-13		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “+” situado en la parte superior derecha 2. El usuario pulsa “Capture video” 3. El usuario realiza un video con la cámara del dispositivo 		
Verificación	<ul style="list-style-type: none"> - El sistema carga el nuevo video - El sistema muestra la vista de recurso para que el usuario añada la información del nuevo recurso 		

Caso de prueba	Nuevo recurso de tipo URI	Identificador	CPR-14
Caso de uso asociado	CU-14		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “+” situado en la parte superior derecha 2. El usuario pulsa “Add new URI” 3. El usuario escribe una nueva dirección URI con el teclado del dispositivo 4. El usuario pulsa el botón “Add” para añadir la URI como un nuevo recurso 		
Verificación	<ul style="list-style-type: none"> - El sistema carga la nueva URI - El sistema muestra la vista de recurso para que el usuario añada la información del nuevo recurso 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Nuevo recurso tipo multimedia del dispositivo	Identificador	CPR-15
Caso de uso asociado	CU-15		
Precondiciones	Se debe haber superado la prueba CPR-04		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario pulsa sobre el botón “+” situado en la parte superior derecha 2. El usuario pulsa “Add new file” 3. El usuario selecciona un archivo multimedia de su dispositivo 		
Verificación	<ul style="list-style-type: none"> - El sistema carga la nueva URI - El sistema muestra el menú de recurso para que el usuario añada la información del nuevo recurso 		

Caso de prueba	Añadir información de nuevo recurso	Identificador	CPR-16
Caso de uso asociado	CU-16		
Precondiciones	Se debe haber superado las pruebas CPR-12, CPR-13, CPR-14 y CPR-15		
Pasos a seguir	<ol style="list-style-type: none"> 1. El usuario escribe con el teclado del dispositivo un nombre al nuevo recurso 2. El usuario escribe con el teclado del dispositivo una descripción al nuevo recurso 		
Verificación	<ul style="list-style-type: none"> - El sistema añade el nuevo recurso a la lista como recurso privado del usuario 		

UNIVERSIDAD CARLOS III DE MADRID

Caso de prueba	Compartir recurso privado	Identificador	CPR-17
Caso de uso asociado	CU-17		
Precondiciones	CPR-16		
Pasos a seguir	1. El usuario pulsa el botón para compartir recurso		
Verificación	- El sistema añade el nuevo recurso a la lista de recursos de todos los usuarios		

Caso de prueba	Modificación de datos de acceso del usuario	Identificador	CPR-18
Caso de uso asociado	CU-18		
Precondiciones	CPR-04		
Pasos a seguir	1. El usuario accede al menú lateral izquierdo 2. El usuario pulsa sobre el botón "Account" 3. El usuario introduce una nueva cuenta de correo electrónico con el teclado de su dispositivo móvil 4. El usuario introduce su contraseña en el campo "Current password" 5. El usuario pulsa el botón "Save"		
Verificación	- El sistema actualiza la información del usuario		

Anexo VI. Prototipo final

En este anexo se presentan las capturas de pantalla del producto final visto desde un dispositivo móvil y una tablet Android.

Prototipo final – móvil Android

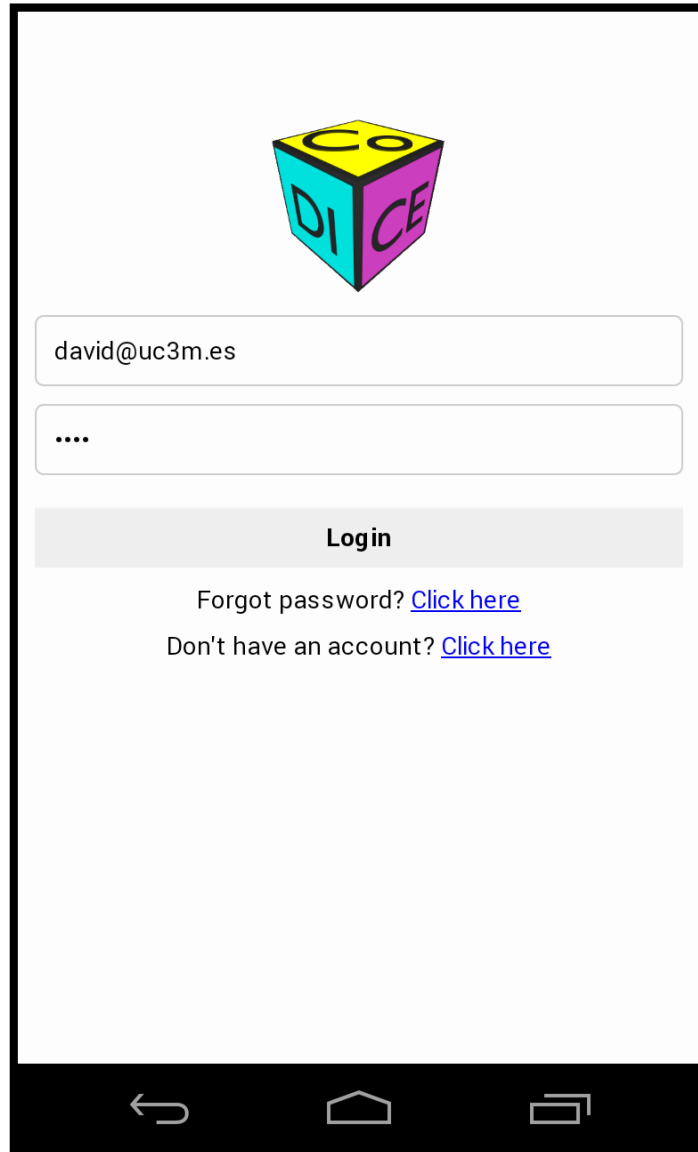



Ilustración 32. Pantalla de login CoDICE versión móvil



Name

Surname

E-mail

Verify E-mail

Password

Confirm password

Sign Up




Ilustración 33. Pantalla de alta de usuario CoDICE versión móvil

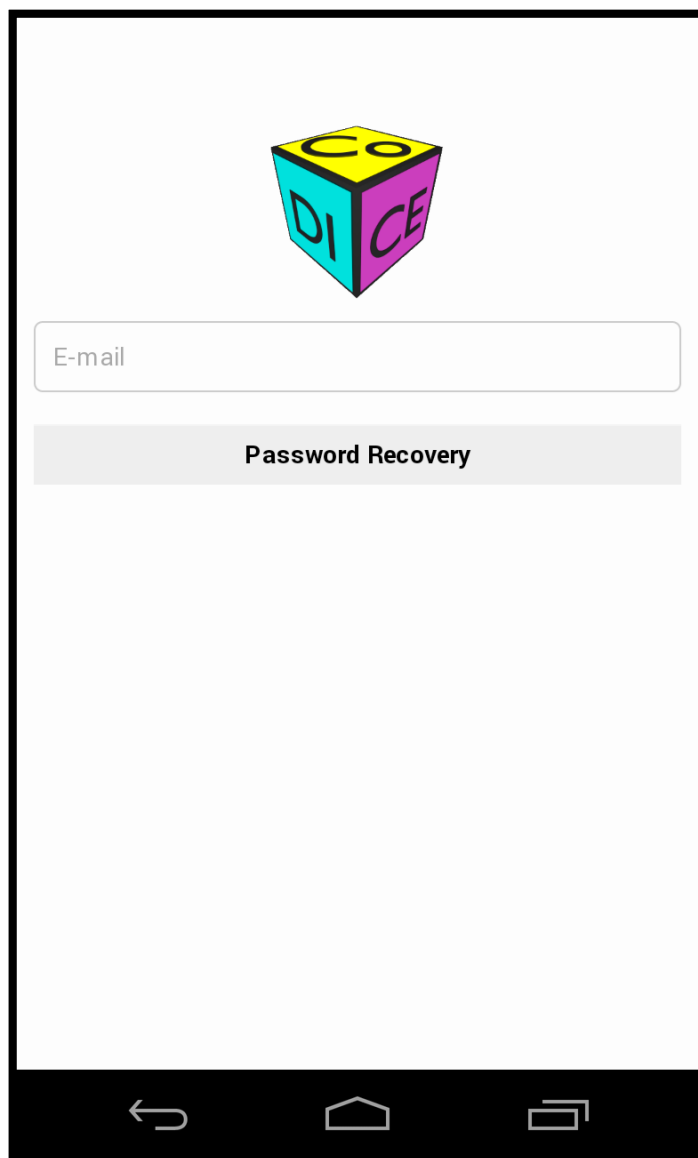


Ilustración 34. Pantalla de recuperación de contraseña CoDICE versión móvil

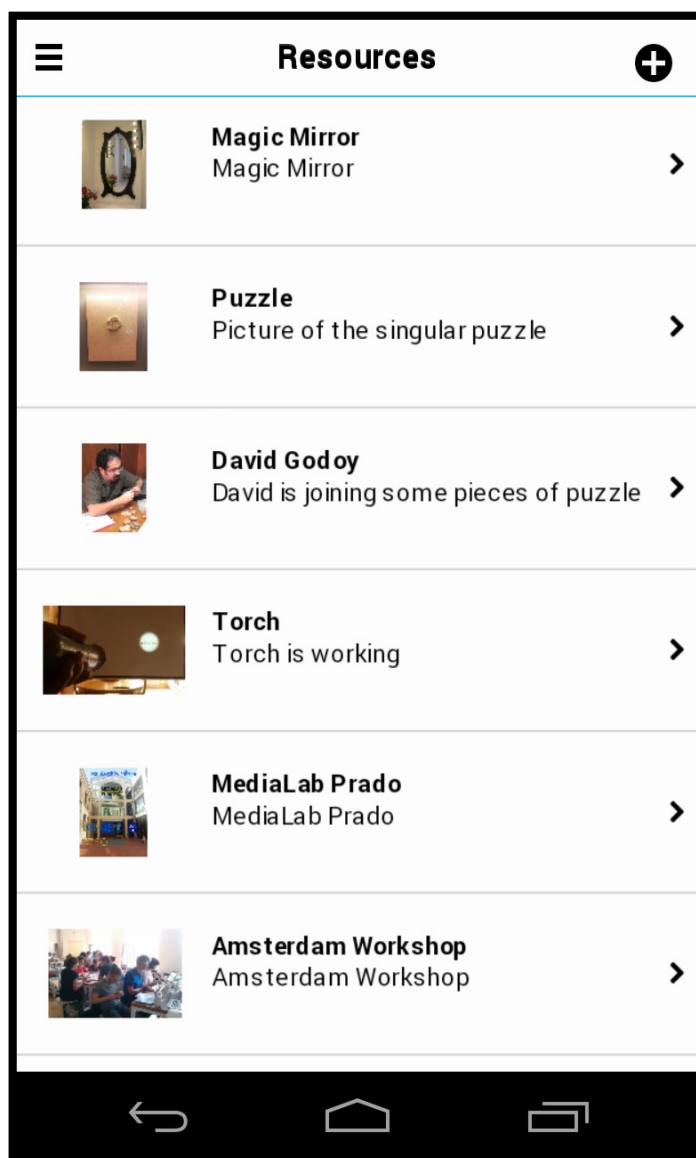


Ilustración 35. Pantalla de recursos CoDICE versión móvil



Ilustración 36. Pantalla de recurso individual CoDICE versión móvil

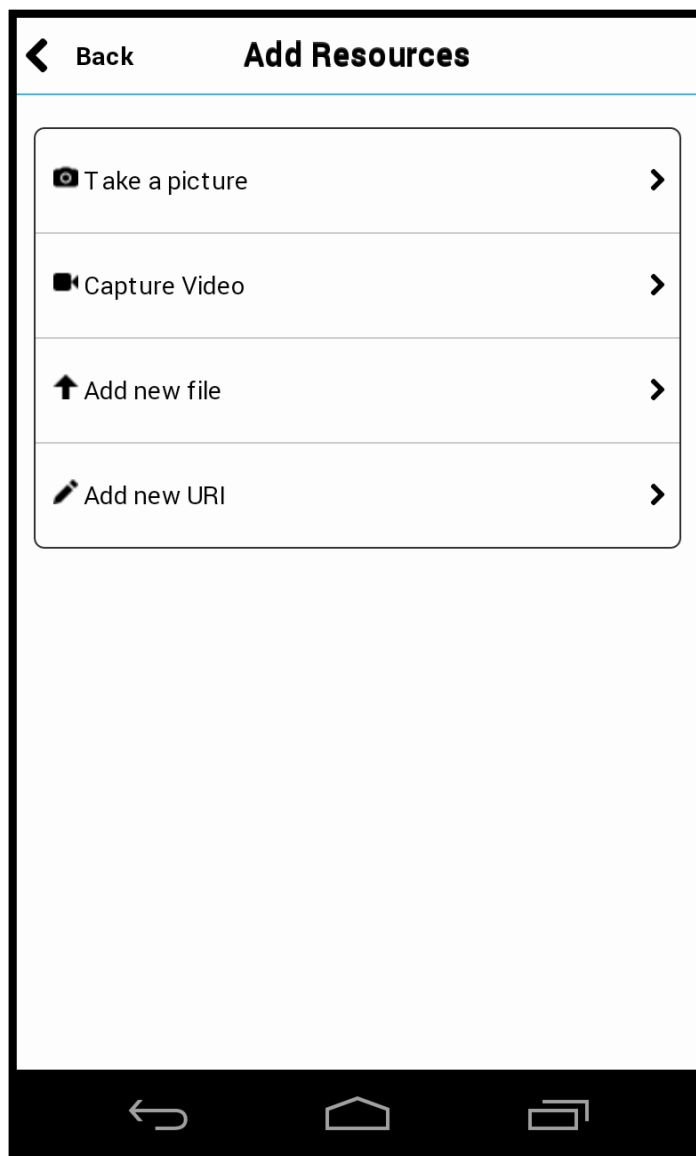


Ilustración 37. Pantalla de menú para añadir recursos CoDICE versión móvil

< Back **My Account**

David

De Leon

david@uc3m.es

.....

Confirm password

To update your account, you have to fill in your current password:

Current password

Save

Ilustración 38. Pantalla de cuenta del usuario CoDICE versión móvil

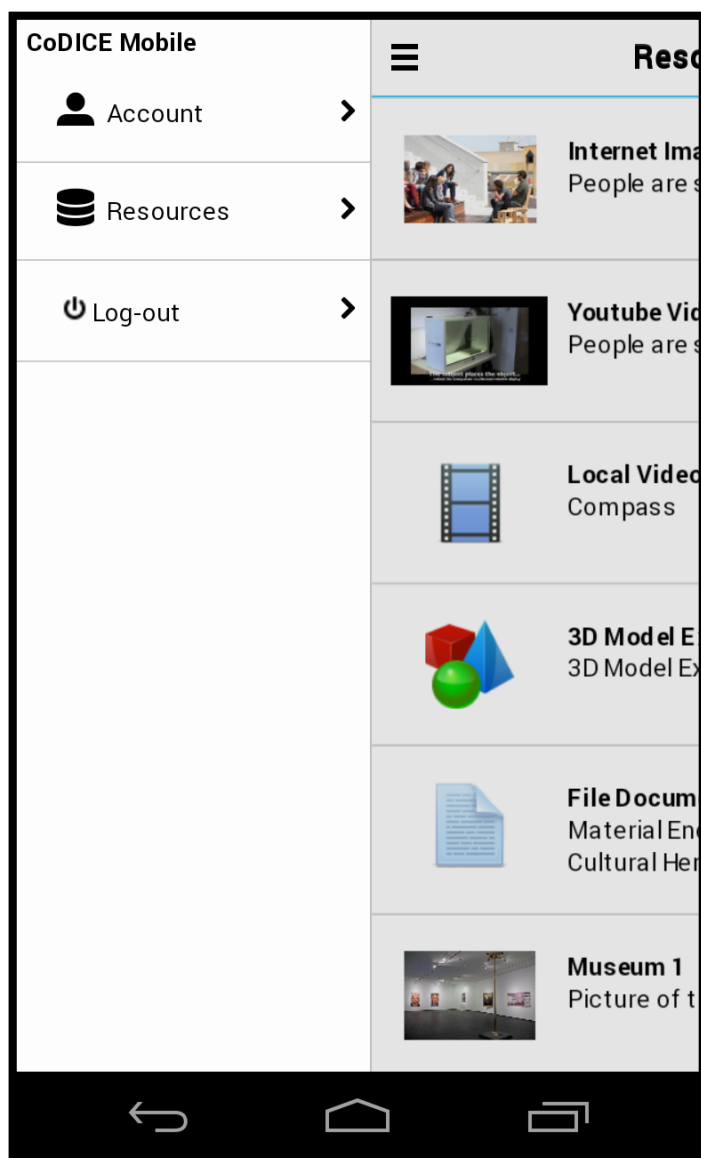


Ilustración 39. Pantalla del menú lateral izquierdo desplegado CoDICE versión móvil

Prototipo final – tablet Android

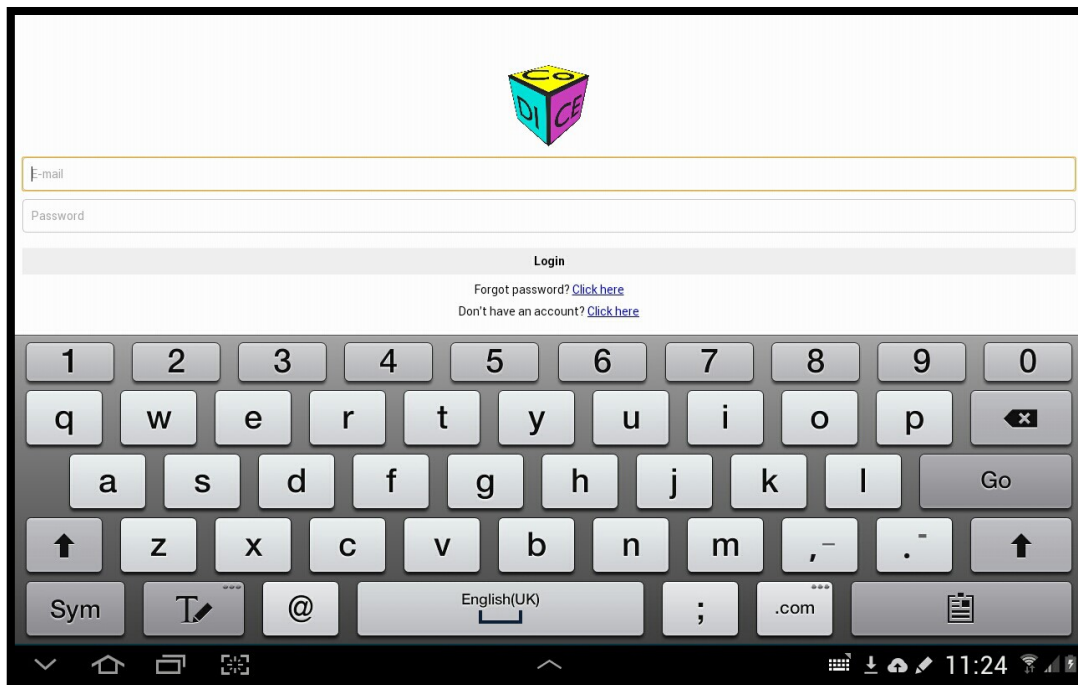


Ilustración 40. Pantalla de login CoDICE versión tablet

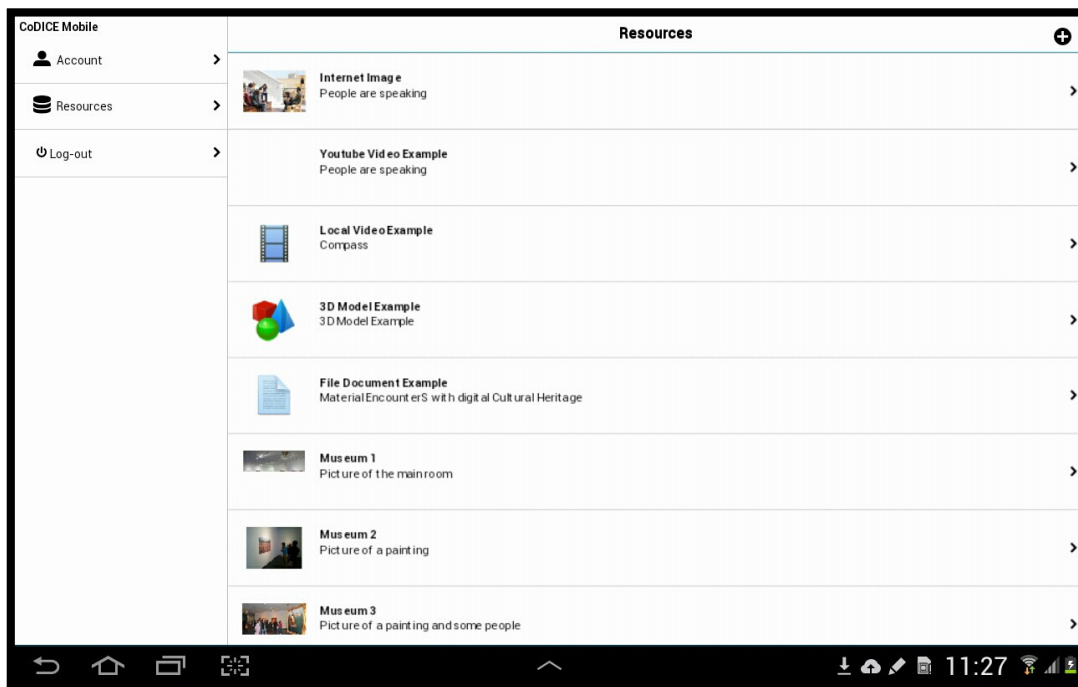


Ilustración 41. Pantalla de recursos CoDICE versión tablet

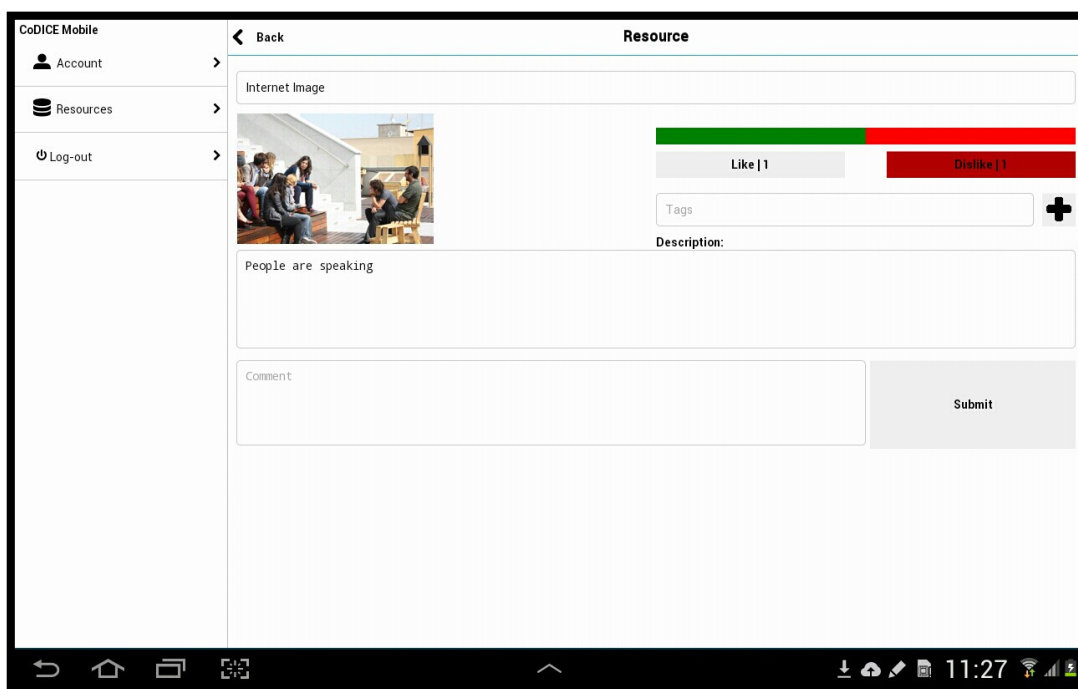


Ilustración 42. Pantalla de recurso CoDICE versión tablet

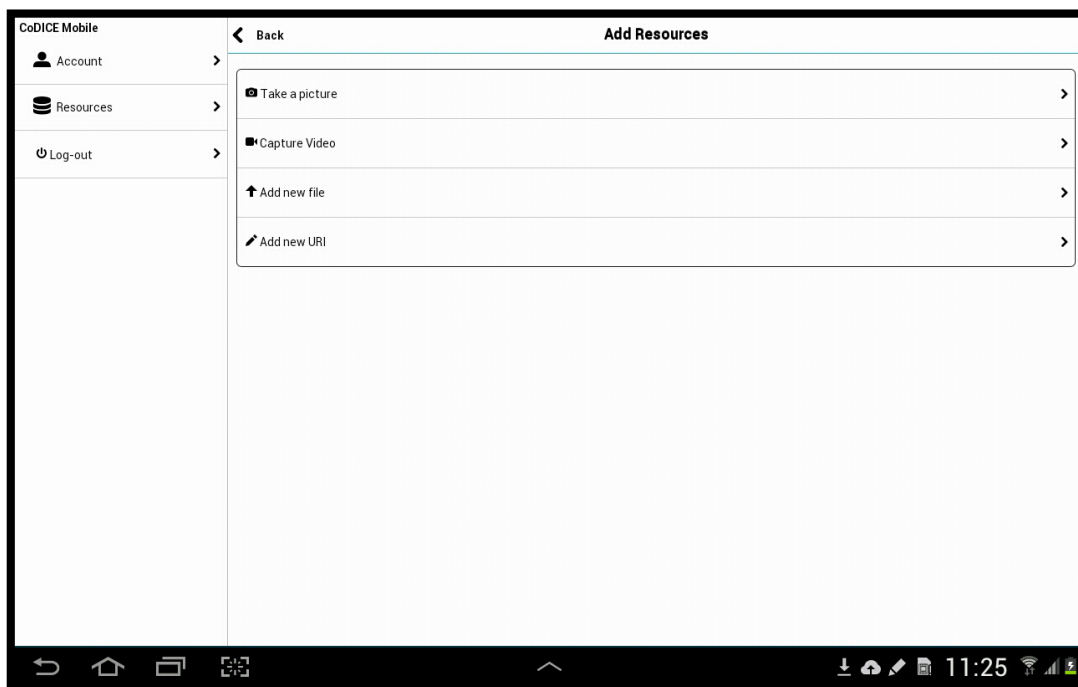


Ilustración 43. Pantalla de añadir recurso CoDICE versión tablet

CoDICE Mobile

Account >

Resources >

Log-out >

< Back

My Account

David

De Leon

david@uc3m.es

.....

Confirm password

To update your account, you have to fill in your current password:

Current password

Save

Ilustración 44. Pantalla cuenta de usuario CoDICE versión tablet